

SUPER

Anno 2
n° 3 Marzo 1985

L. 7.500

Spedizione in
abbonamento
postale Gruppo III/70

3

COMMODORE

Una pubblicazione della

J.soft EDITRICE

in collaborazione con

GRUPPO
EDITORIALE
JACKSON



SUPERCOMMODORE
la rivista per gli utenti
dei prodotti COMMODORE

SUPERTURBO PER VIC 20 E C 64

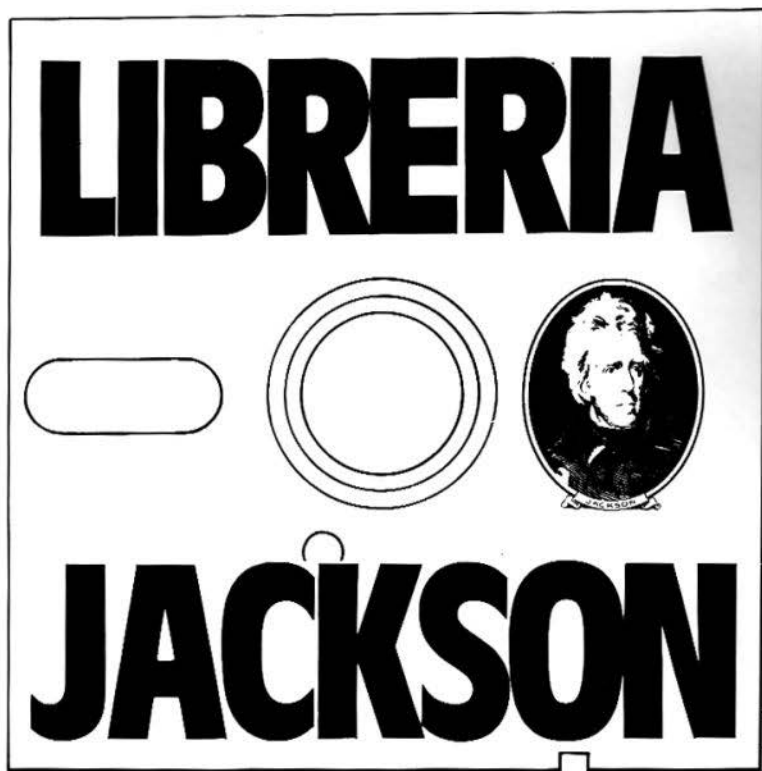
DEFLECTOR PER VIC 20

TRAFIC PER C 64

LABIRINTO 3D

**ROULETTE PER
C 16**





**A Milano, in via Mascheroni, 14.
La prima software - libreria italiana**

Un tempo si andava in libreria per il gusto della scoperta, per il piacere di esser informati sulle novità. Per incontrarsi, discutere, chiedere un consiglio al libraio-amico. Tutto questo è ancora possibile, per un prodotto assolutamente nuovo: libri e riviste di informatica italiani ed esteri, software, giochi.

Dove? Alla **Libreria JACKSON**. La prima software - libreria italiana.

A Milano, in via Mascheroni, 14. - Tel. 02-437385

Vieni a trovarci: ti aspettiamo.

SUPER

COMMODORE

5 EDITORIALE POINT

GIRA, GIRA...
di **Pietro Dell'Orco**

7 READ & WRITE

LA POSTA DEI LETTORI

11 OPEN

**UNA PRIMA OCCHIATA
AL COMMODORE C16**
di **M. Cristuib Grizzi**

14 LA CULTURA IS INFORMATICA then

**COME SFRUTTARE LA
TECNICA DI
MEMORIZZAZIONE DEI
PROGRAMMI NEL VIC
20 E NEL C64**
di **A. Borra**

20 REM:HW

IL JOYSTICK
Parte seconda
di **U. G. Barzaghi**

29 LOAD

**29 SUPERTURBO PER
VIC 20 E C64**
di **H. De Cevkelaire**
trad. e adatt.
di **M. Cristuib Grizzi**

**35 NEWSLETTER
EDITOR**
di **F. Thompson**
trad. e adatt.
di **U. Barzaghi**

44 LABIRINTO 3D
di **K.M. Queen**
trad. e adatt.
di **F. Sarcina**

47 REVERSI
di **F. Puckett**
trad. e adatt.
di **F. Sarcina**

**54 CARATTERI
GIGANTI
PER VIC 20 E C64**
di **T. Heimark**
trad. e adatt.
di **F. Sarcina**

**57 DEFLECTOR PER
VIC 20**
di **S.J. Tyniw**
trad. e adatt.
di **S. Anticoli e
M. Albarelli**

60 TRAFIC PER C64
di **M. Cello Zanollo**

**63 ROULETTE PER
C16**
di **M. Cristuib Grizzi**

**69 MAGIC DRAW PER
VIC 20**
di **K. Gough**
trad. e adatt.
di **M. Cristuib Grizzi**

72 VIC-sound

**PRINT SOUND PER
VIC 20**
di **A. Kapuan**
trad. e adatt.
di **F. Sarcina**

77 RUN

**RECENSIONI
SOFTWARE**
di **M. Cristuib Grizzi**

78 PRINT

RECENSIONI LIBRI
di **M. Cristuib Grizzi**

80 Input-output

PICCOLI ANNUNCI

J. soft s.r.l.
**DIREZIONE, REDAZIONE,
AMMINISTRAZIONE**
Viale Restelli, 5
20124 MILANO
Tel. (02) 68.88.228

DIRETTORE RESPONSABILE:
Pietro Dell'Orco

COORDINAMENTO TECNICO:
Riccardo Paoillo

REDAZIONE:
Mauro Cristuib Grizzi

**HANNO COLLABORATO A
QUESTO NUMERO**
S. Anticoli
M. Albarelli
U.G. Barzaghi
A. Bozza
F. Sarcina
M. Zanollo

GRAFICA E IMPAGINAZIONE:
Margherita La Noce
Ivana Rossi
Raffaella Tofolatti

FOTOCOPOSIZIONE:
Graphotek - Via Artesani, 16
Tel. 02/64.80.397
20161 MILANO

CONTABILITÀ:
Giulia Pedrazzini
Flavia Bonatti

**AUTORIZZAZIONE ALLA
PUBBLICAZIONE:**
Tribunale di Milano n° 201
del 14.04.1984

STAMPA:
Litografia del Sole
Albairate (MI)

PUBBLICITÀ
Concessionaria per l'Italia e
l'Estero J. Advertising s.r.l.
Viale Restelli, 5
20124 MILANO
Tel. (02) 68.82.895-68.80.606-68.87.233
Tlx. 316213 REINA I

Concessionaria esclusiva per la
DIFFUSIONE in Italia e Estero:
SODIP - Via Zuretti, 25
20125 MILANO

Spedizione in abbonamento
postale Gruppo III/70
Prezzo della rivista L. 7.500
Numero arretrato L. 15.000
Abbonamento annuo: L. 82.500
(11 numeri con cassetta);
L. 38.500 (11 numeri senza cassetta);
per l'estero: L. 110.000 (11 numeri con
cassetta) - L. 50.000 (11 numeri senza
cassetta)

I versamenti devono essere indirizzati a:
J. soft s.r.l.
Viale Restelli, 5 - 20124 MILANO
mediante emissione di
assegno bancario, cartolina
vaglia o utilizzando
il c/c postale n. 19445204.
Per i cambi di indirizzo indicare,
oltre al nuovo, anche l'indirizzo
precedente ed allegare alla
comunicazione l'importo di L. 500,
anche in francobolli

© TUTTI I DIRITTI DI
RIPRODUZIONE O TRADUZIONE
DEGLI ARTICOLI PUBBLICATI
SONO RISERVATI

GLI ARTICOLI TRADOTTI SONO
TRATTI DALLA RIVISTA COMPUTE!
O DA COMPUTE! GAZZETTE
COMPUTE! PUBLICATION, INC.
P.O. BOX 5406,
GREENSBORO, NC 27403 - USA

LIBRI FIRMATI JACKSON

Rita Bonelli
Luciano Pazzucconi
Fabio Racchi
Giovanni Valerio

Commodore 64 la grafica e il suono

Ogni argomento viene spiegato e accompagnato da numerosi esempi commentati. Nel libro sono listati moltissimi programmi che sono registrati sulla cassetta allegata. Nel Capitolo 1 si tratta del video, della tastiera e della grafica in modo caratteristico. Il Capitolo 2 è dedicato alla grafica e ne approfondisce tutti gli aspetti. Il Capitolo 3 tratta degli sprite. Nel Capitolo 4 viene trattato l'argomento del suono. Completano il libro tre appendici: i registri del VIC II e del SID e le frequenze delle note musicali.

Cod. 409 B Pag. 270 Lire 34.000

Rita Bonelli
Commodore 64
i file

Questo libro tratta in maniera completa e precisa la gestione dei file su cassetta e su disco. Oltre a brevi programmi esempio, riportati per spiegare l'uso delle istruzioni, il libro contiene cinque programmi per creare e gestire un archivio di dati: SEQUENZIALE su cassetta, SEQUENZIALE su disco, RANDOM su disco, RANDOM/USER su disco, RELATIVO su disco.

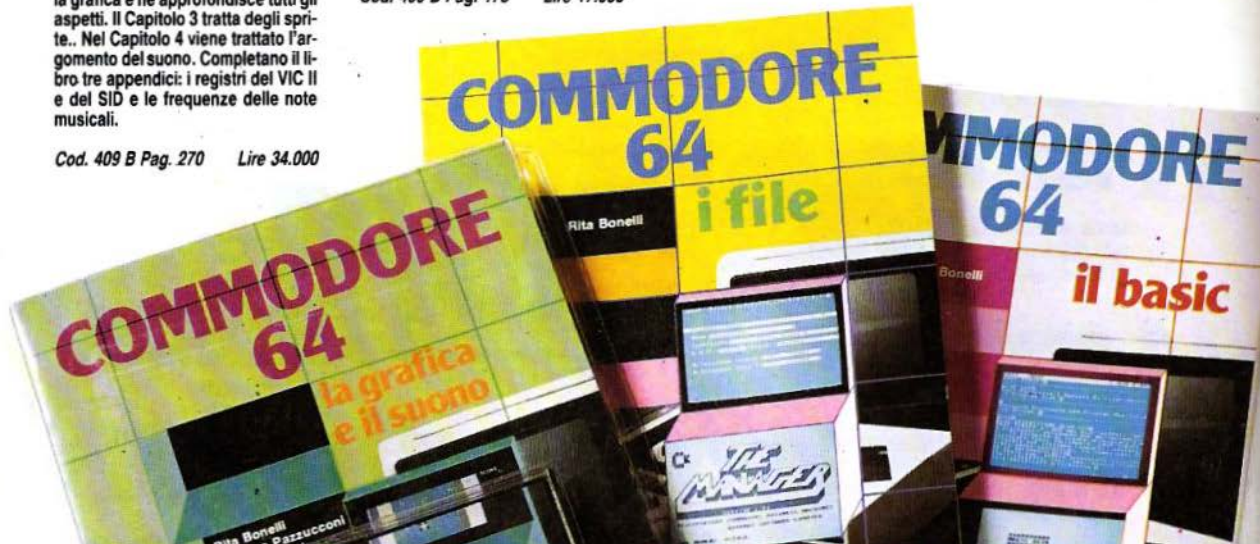
Cod. 400 B Pag. 173 Lire 17.000

Rita Bonelli
Commodore 64
il basic

Un'accurata esposizione del linguaggio BASIC, accompagnata da numerosi esempi. Nel Capitolo 1 si ha una panoramica dei diversi argomenti. Il Capitolo 2 è dedicato al linguaggio. Nel Capitolo 3 si approfondisce l'uso della tastiera e del video. Il Capitolo 4 fornisce le informazioni necessarie per usare disco e cassetta per memorizzare programmi.

Il Capitolo 5 è dedicato alla stampante. Nel Capitolo 6 si parla della costruzione del programma. Nel Capitolo 7 vengono passati in rassegna i codici e i numeri del calcolatore. Il Capitolo 8 è dedicato alla memoria. Nel Capitolo 9 si tratta degli errori. Completano il libro, l'Appendice A dedicata alla tastiera e l'Appendice B all'argomento del BASIC compilato. Tutti i programmi esempio riportati nel libro sono disponibili a richiesta su floppy disk.

Cod. 348 D Pag. 316 Lire 26.000



LA BIBLIOTECA CHE FA TESTO.

ritagliare (o fotocopiare) e spedire in busta chiusa a:
GRUPPO EDITORIALE JACKSON - Divisione Libri - Via Rosellini, 12 - 20124 Milano

CEDOLA DI COMMISSIONE LIBRARIA

VOGLIATE SPEDIRMI			
n° copie	codice	Prezzo unitario	Prezzo totale
Totale			

Pagherò contrassegno al postino il prezzo indicato più L. 3000 per contributo fisso spese di spedizione.

Condizioni di pagamento con esenzione del contributo spese di spedizione:

Allego assegno della Banca Allego fotocopia del versamento su c/c n. 11866203 a voi intestato

N° _____ Allego fotocopia di versamento su vaglia postale a voi intestato

Nome _____
 Cognome _____
 Via _____
 Cap _____ Città _____ Prov. _____
 Data _____ Firma _____

Spazio riservato alle Aziende. Si richiede l'emissione di fattura ORDINE MINIMO L. 50.000

Partita I.V.A. _____

POINT

Gira, Gira...

Riceviamo numerose lettere di lettori che protestano per il mancato funzionamento dei programmi pubblicati sulla rivista. Parliamone. Prima della pubblicazione ogni programma viene accuratamente e a lungo provato, testato e giocato (se di gioco si tratta) in redazione, proprio per controllarne il corretto funzionamento.

La produzione del listato e della cassetta avviene utilizzando poi un master "supercollaudato" (per una super rivista è necessario un supercollaudo). Come abbiamo già avuto modo di dire, può succedere che, per motivi di carattere tecnico e del tutto imprevedibili, qualche cassetta risulti non perfettamente funzionante. Tutto ciò rientra, nostro malgrado, nella normalità delle cose.

Il medesimo discorso vale per la pubblicazione dei listati; ogni tanto qualcosa può sfuggire agli occhi attenti di redattori, grafici, ecc.

In ogni caso si tratta sempre e comunque di eventi con carattere di eccezione. Quando abbiamo avuto modo di controllare le asserzioni di certi lettori che sostenevano il malfunzionamento di un programma è sempre risultato un errore di caricamento da cassetta oppure un errore di trascrizione dello stesso.

Nel primo caso non erano state lette, e correttamente interpretate, le istruzioni per il caricamento da nastro; nel secondo caso, nonostante l'OK List, c'era un lettore di trascrizione nel programma.

I più frequenti tra questi errori sono lo scambio dello 0 (zero) con la lettera "O" oppure la errore "I" con la cifra 1 e così via.

Quanto detto non vuole certo essere una difesa ad oltranza del nostro lavoro, ma non è giusto che ci vengano addossate colpe per lo più inesistenti.

Vi chiediamo dunque di porre un po' più di attenzione, poiché siamo certi che, così facendo, potranno essere risolti la maggior parte dei problemi circa il corretto funzionamento di cassette e programmi.

Pietro Dell'Orco

Per collaborare a SUPERCOMMODORE

La rivista è interessata ad articoli e programmi riguardanti la linea Commodore, di interesse generale, chiari ed esaurienti. Vi proponiamo una piccola "guida", che ha particolarmente lo scopo di rendere più accurata la stesura dei vostri manoscritti: seguendo i nostri consigli si accresceranno le probabilità che le vostre buone idee e i vostri programmi vengano pubblicati.

1 l'angolo superiore sinistro della prima pagina dovrà contenere: nome, cognome, indirizzo, numero telefonico, codice fiscale, data di spedizione, luogo e data di nascita.

2 l'angolo superiore destro della prima pagina dovrà contenere la marca e il tipo di computer al quale il lavoro si riferisce, unitamente ad eventuali espansioni di memoria o periferiche richieste.

3 il titolo dell'articolo, sottolineato, dovrà iniziare a circa due terzi in altezza della prima pagina.

4 le pagine seguenti potranno essere battute normalmente, con la condizione che l'angolo superiore destro contenga un'abbreviazione del titolo e del cognome, unitamente al numero di pagina. Per esempio, Sprite Ed.../Brambilla/2.

5 il testo dell'articolo dovrà essere battuto a macchina con interlinea di uno spazio e mezzo, massimo due spazi; un margine di almeno un centimetro dovrà essere lasciato su entrambi i lati dello scritto.

6 dovranno essere usati fogli in formato UNI A4 (cm. 21 x 29,7) e il testo, scritto in caratteri maiuscoli e minuscoli, dovrà occupare una sola facciata del foglio.

7 nel caso il testo comprenda più fogli, questi dovranno essere uniti con un fermaglio o con un punto metallico aperto.

8 avendo intenzione di spedire più di un articolo, questi dovranno essere inviati separatamente insieme alla rispettiva copia su supporto magnetico.

9 programmi brevi (meno di 20 linee) potranno essere inseriti nel testo, mentre programmi più lunghi dovranno essere listati separatamente. È **ESSENZIALE** per noi disporre di una copia del programma registrata più volte su supporto magnetico, su entrambi i lati dello stesso. È preferibile usare nastri di buona qualità e di lunghezza non eccessiva; la cassetta o il disco dovranno essere etichettati con il nome dell'autore, il titolo dell'articolo, il computer interessato e soprattutto le eventuali espansioni richieste. Come sug-

gerimenti di programmazione si consiglia di usare le istruzioni "CHR\$(x)", "TAB(x)", "SPC(x)", piuttosto che stringhe di manipolazione del cursore.

Ad esempio, per uno scroll di 5 linee l'istruzione "FORI = 1 TO 5:PRINT:NEXT". è molto più interpretabile di 5 Q inverse; e, invece di una dozzina di simboli di cursore a destra, perché non usare semplicemente "PRINT SPC (12)"? Un rapido controllo dei programmi per operare queste sostituzioni sarà molto apprezzato da noi e dai lettori.

10 per maggior chiarezza, all'interno dell'articolo è conveniente usare caratteri maiuscoli riferendosi a istruzioni BASIC (esempio RETURN, LIST, RND, PRINT ecc.). Se si desidera evidenziare una parola, è preferibile sottolinearla piuttosto che scriverla in carattere maiuscolo.

11 gli articoli ed i programmi potranno avere qualsiasi lunghezza: da una routine di una sola linea fino a programmi molto complessi.

12 volendo includere diapositive, queste dovranno avere formato 24x36, o 6x6.

13 non prenderemo in considerazione articoli che siano stati sottoposti ad altre case editrici.

14 il materiale non pubblicato non verrà restituito.

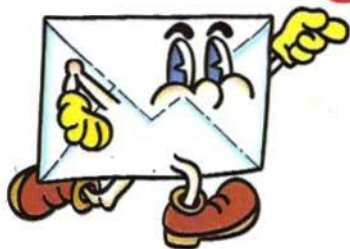
15 il compenso per la collaborazione prestata sarà commisurato alla complessità e all'interesse del testo e/o del programma (da un minimo di L. 50.000 a un massimo di L. 300.000). Il pagamento verrà effettuato in caso di pubblicazione del lavoro.

Inviare idee e programmi a:
SUPERCOMMODORE
Viale Restelli, 5
20124 Milano

e saremo lietissimi di pubblicare i contributi migliori.

La Redazione

READ & WRITE



La posta
dei lettori

Come funziona il Checksum di MLX

Spett. Redazione,

sono una vostra abbonata fin dal primo numero, ed io e mio fratello seguiamo la rivista con interesse perché contiene programmi facili da copiare grazie ad OK-LIST ed MLX.

A proposito di quest'ultimo, non capisco come viene calcolato il valore di checksum nella linea 500, e questo mi servirebbe per fare delle prove. Potete spiegarcelo e fare l'esempio con una riga di programma?

Licia Canal

R. MLX calcola il checksum dopo che sono stati inseriti i primi sei numeri di una linea e lo confronta con il settimo: se è uguale, passa alla linea successiva, mentre se è diverso stampa un messaggio d'errore e richiede che venga ridigitata la linea errata.

Per ogni numero di tre cifre digitato MLX incrementa una variabile - chiamiamola N - di un valore dipendente dai codici ASCII dei tre numeri digitati, secondo la formula:

$$N=N \cdot 10+A-48$$

dove A è il codice ASCII della singola cifra componente il numero. Questo valore viene poi immagazzinato in un vettore di 6 elementi - chiamiamolo V(6). Il checksum - chiamiamolo CK - viene inizialmente calcolato sull'indirizzo della linea, secondo la formula:

$$CK=I-INT(I/256)*256$$

dove I è l'indirizzo della linea. Un ciclo che si ripete sei volte incrementa il checksum dal valore così calcolato ad un valore finale dato dalla formula:

$$CK=(CK+A(J))AND255$$

dove J è la variabile del ciclo.

Facciamo un esempio pratico su una linea di MLX:

49152:011,008,000,000,158,050,227
i codici ASCII di 011 sono 48 49 49
i codici ASCII di 008 sono 48 48 56
i codici ASCII di 000 sono 48 48 48
i codici ASCII di 000 sono 48 48 48
i codici ASCII di 158 sono 49 53 56
i codici ASCII di 050 sono 48 53 48

Inizialmente N varrà 0. Il confronto sul primo numero darà:

$$N=0 \cdot 10+48-48=0$$
$$N=0 \cdot 10+49-48=1$$
$$N=1 \cdot 10+49-48=11$$

Il vettore A per il primo elemento conterrà allora il valore 11: A(0)=11. N viene posto nuovamente a zero, e si passa al secondo numero:

$$N=0 \cdot 10+48-48=0$$
$$N=0 \cdot 10+48-48=0$$
$$N=0 \cdot 10+56-48=8$$

Sarà così A(1)=8. Procedendo così per gli altri quattro numeri si ottengono i valori A(2)=0, A(3)=0, A(4)=158, A(5)=50.

Viene ora calcolato il checksum per l'indirizzo:

$$CK=49152-INT(49152/256)*256=0$$

Verranno poi effettuati gli incrementi sulla base del contenuto del vettore A(J):

$$CK=(0+11)AND255=11$$
$$CK=(11+8)AND255=19$$
$$CK=(19+0)AND255=19$$
$$CK=(19+0)AND255=19$$
$$CK=(19+158)AND255=177$$
$$CK=(177+50)AND255=227$$

Come si vede, il checksum finale risulta 227, corrispondente all'ultimo numero

della linea. Questi calcoli, laboriosi se fatti a mano, vengono effettuati in tempo reale dal computer ogni volta che si digita una linea con MLX. Per produrre i listati in formato MLX noi disponiamo di un programma che opera il procedimento inverso: caricato in memoria un programma in linguaggio macchina il nostro "MLX Inverso" legge con una serie di PEEK il contenuto delle locazioni interessate, e per ogni sei locazioni calcola il checksum, che viene poi stampato come settimo numero nel listato. Ecco come, con un procedimento complicato, si rende semplice la digitazione di un listato.

Emulare il C16

Spett. Redazione,

so che il C16 possiede un BASIC molto più completo di quello disponibile sul C64, e pensavo alla possibilità di salvare su disco l'intero contenuto della memoria del C16 dalla prima locazione all'ultima (compreso quindi Interprete BASIC e Sistema Operativo), quindi con la selezione a banchi rendere "pura RAM" il mio C64 e caricarvi la memoria del C16. Credete sia possibile?

Marco Chierichetti

R. Il C16 ed il C64 presentano differenze hardware, oltre che software, tali da rendere pressoché impossibile questa operazione. Oltre a ciò, occorrerebbe scrivere un'apposita routine in linguaggio macchina che provvedesse al caricamento della memoria del C16 nel C64, dopo che quest'ultimo fosse stato reso "pura RAM", dal momento che sarebbero state disabilitate anche tutte le routine di I/O del Kernel del C64. Inoltre, la routine dovrebbe passare il controllo al Sistema Operativo del C16 una volta che questo fosse stato caricato in RAM.

Piccola guida per l'input dei programmi

Molti dei programmi listati da **SUPERCOMMODORE** contengono particolari caratteri di controllo (controllo cursore, tasti colore, video-inverse, ecc.). Per rendere chiaro ciò che dev'essere battuto quando è necessario inserire uno dei suddetti caratteri sono state stabilite le seguenti convenzioni.

Generalmente i listati per VIC 20 e C64 contengono delle «parole» racchiuse tra parentesi graffe {}; tali parole rappresentano particolari caratteri di controllo: {GIU'} significa premere il simbolo del cursore verso il basso. {5 SPAZI} vuol dire battere 5 volte la barra-spazio.

Per indicare che un tasto dev'essere «shiftato», cioè premuto insieme al tasto SHIFT, nel listato il simbolo di questo tasto sarà sottolineato.

Per esempio S significa che dev'essere premuto il tasto S mentre è abbassato il tasto SHIFT.

La manovra farà apparire sullo schermo un piccolo «cuore».

Trovando un simbolo sottolineato chiuso tra parentesi graffe (esempio {10N}) dovrà essere interpretato come «premere il carattere indicato per il numero di volte che lo precede nelle parentesi» nel nostro esempio premere 10 volte il tasto N «shiftato».

Se il tasto da premere è indicato tra le parentesi [()] vuol dire che lo stesso dovrà essere premuto mentre è premuto il tasto «Commodore» (il tasto «Commodore» è quello posto nell'angolo in basso a sinistra).

Se il tasto scritto tra [()] è preceduto da un numero, ciò significa che il tasto dev'essere premuto per il numero di volte indicato.

Raramente si potrà trovare un carattere alfabetico solitario racchiuso tra parentesi graffe. Tale carattere sul C64 può essere battuto mentre è premuto il tasto CTRL.

Ad esempio {A} sta ad indicare la sequenza CTRL-A. A proposito del «modo virgolette» è nota la possibilità di muovere il cursore sullo schermo con i tasti CRSR. Ogni tanto i programmatori desiderano muovere il cursore durante l'esecuzione del programma. È per questo motivo che nei programmi si troveranno dei {SIN}, {HOME} e {BLU}.

L'unico modo perché il calcolatore distingua il comando diretto da quello inserito nel programma è il «modo virgolette». Premendo il tasto «virgolette» (tasti SHIFT 2) il calcolatore si predispose appunto in «modo virgolette».

Battendo un carattere qualsiasi cercando poi di correggerlo muovendo il cursore, si otterrà solo un tratto verticale in un quadratino inverso. Questo è infatti il simbolo del cursore a sinistra nel «modo virgolette». L'unico comando di editing non utilizzabile all'interno di un programma è il DEL. Battendo nuovamente le «virgolette» il calcolatore lascia il «modo virgolette». Si può accedere al «modo virgolette» quando inserite spazi nella linea.

In ogni caso, il sistema più semplice per uscire dal «modo virgolette» è quello di premere il tasto RETURN.

Utilizzare la tabella che segue quando è necessario inserire comandi relativi al cursore e ai tasti colore.

Quando leggete:	Premete:	Vedrete:	Quando leggete:	Premete:	Vedrete:	Quando leggete:	Premete:	Vedrete:
{CLR}	SHIFT CLR/HOME		{CYN}	CTRL 4		[<7>]		
{HOME}	CLR/HOME		{PUR}	CTRL 5		[<8>]		
{SU}	SHIFT ↑ CRSR ↓		{GRN}	CTRL 6		{F1}		
{GIU'}	↓ CRSR ↓		{BLU}	CTRL 7		{F2}		
{SIN}	SHIFT ⇐ CRSR ⇒		{YEL}	CTRL 8		{F3}		
{DES}	⇐ CRSR ⇒		[<1>]			{F4}		
{RVS}	CTRL 9		[<2>]			{F5}		
{OFF}	CTRL 0		[<3>]			{F6}		
{BLK}	CTRL 1		[<4>]			{F7}		
{WHT}	CTRL 2		[<5>]			{F8}		
{RED}	CTRL 3		[<6>]					

SUPERCAR.



180 km/h – 9.9 sec da 0 a 100 km/h

La tentazione è forte. 96 CV pronti su ogni percorso, 180 km/h, 9.9 secondi da 0 a 100 km/h. Brillante, sportiva, decisa. Basta guardarla. Carreggiata larga, pneumatici radiali a profilo ribassato 185/60 HRT3 su cerchi in lega da 6". Profilo aggressivo, aderenza perfetta. Basta guidarla.

XR2 è tecnologia Ford. Trazione anteriore, propulsore 1.6 CVH, 4 cilindri monoalbero a camme in testa, 5 marce, freni anteriori a disco autoventilanti, sospensioni rinforzate con barra stabilizzatrice posteriore.

XR2. Dura, selvaggia, anzi dolcissima. Basta sedersi. Sedili anteriori rigorosamente ergonomici rivestiti con tessuti esclusivi, sedile posteriore a ribaltamento frazionato, strumentazione e volante sportivi, orologio digitale multifunzione, fari ausiliari di profondità, vetri atermici, apertura elettrica del bagagliaio.

XR2. Al di sopra delle parti.

Tutte le vetture Ford sono coperte da garanzia 1-3-6 (un anno di garanzia estensibile a tre con "La Lunga Protezione" e sei anni di garanzia contro la corrosione perforante) e assistite in oltre 1000 punti di servizio. Finanziamenti Ford Credit e cessioni in Leasing.

FIESTA XR2

Tecnologia e temperamento.



La piccola guida del principiante

Che cos'è un programma?

Il solo computer non può compiere alcuna operazione. Un computer possiede potenzialità, ma, come una macchina senza benzina, senza un programma non può funzionare. La maggior parte dei programmi pubblicati su **SUPERCOMMODORE** per i computer Commodore sono scritti in un linguaggio per calcolatori chiamato BASIC. Il BASIC è facile da imparare ed è disponibile, di serie, nel VIC 20 e nel Commodore 64.

Programmi BASIC

Ogni mese **SUPERCOMMODORE** pubblica programmi sia per il VIC 20 che per il C64. Tanto per cominciare, se possiedi un VIC 20 copia solo i programmi scritti per la tua macchina, indicati con "versione per VIC 20". Più tardi, quando avrai acquisito esperienza con il BASIC del tuo computer, potrai cercare di copiare e convertire programmi scritti per altri calcolatori. Diversamente dal linguaggio corrente, che può essere variamente interpretato, il BASIC di solito ha un solo modo corretto per indicare qualcosa. Ogni lettera, carattere o numero ha il suo significato.

Un errore banale è costituito dalla sostituzione del numero "0" con la lettera "O" oppure il carattere minuscolo "l" invece del numero "1" o ancora il carattere maiuscolo "B" con il numero "8".

Devi anche inserire tutta la punteggiatura, i due punti (:) e le virgole, copiando esattamente ciò che appare sulla rivista. Gli spazi possono essere importanti. Per essere sicuro **copia il listato esattamente** come si presenta.

Le parentesi e i caratteri speciali

L'eccezione per questa regola di copiatura si presenta quando incontrerai indicazioni tra parentesi, quali: "{GIU}". Ogni cosa compresa tra parentesi è un carattere speciale oppure un carattere che non può essere facilmente prodotto con la stampante. Incontrando un carattere di questo tipo fai riferimento alla "Piccola guida per l'input dei programmi".

Le istruzioni DATA

Alcuni programmi contengono una sezione, o delle sezioni, di istruzioni DATA. Queste linee di istruzione forniscono le informazioni di cui il programma ha bisogno. In alcuni casi le istruzioni DATA costituiscono il programma vero e proprio, in altri contengono codici grafici. Queste

linee sono particolarmente soggette agli errori. Se un solo numero in una linea di istruzione DATA è sbagliato, il calcolatore potrebbe "piantarsi" o distruggere il programma. La tastiera e il tasto STOP appaiono inattivi e lo schermo completamente vuoto. Non lasciarti prendere dal panico. Non si è verificato alcun danno.

Per riprendere il controllo devi spegnere il computer e successivamente riaccenderlo. Ciò cancellerà qualsiasi programma presente in memoria, per cui è sempre necessario fare il SAVE del tuo programma prima di comandare il RUN.

Se il computer si ferma, puoi caricare (LOAD) il programma e cercare l'errore.

A volte, quando il programma viene "lanciato", un'istruzione DATA errata può causare un messaggio di errore.

Il messaggio di errore potrebbe riferirsi alla linea di programma che legge (READ) il contenuto delle istruzioni DATA.

Come conoscere il computer

Dovresti prendere confidenza con il computer prima di procedere alla copiatura del programma.

Impara le istruzioni che si usano per memorizzare e richiamare i programmi da nastro o da disco. Dovrai conservare una copia del tuo programma, se non vorrai copiarlo ogni volta che lo devi usare. Impara ad usare le funzioni di "editing" della tua macchina. Come puoi correggere un errore? Puoi sempre ricopiare la linea e in questo caso devi sapere come procedere. Sapresti come inserire i caratteri in "inverse", i caratteri minuscoli e quelli di controllo?

Tutto ciò è spiegato nel manuale del calcolatore.

Un veloce ripasso

- 1) Copia il programma una linea alla volta, con ordine. Premi RETURN alla fine di ogni linea. Usa il tasto "DEL" per correggere gli errori.
- 2) Confronta la linea copiata con quella presente nella rivista. Puoi controllare l'intero programma nel caso in cui si presenti un errore quando esegui il RUN.
- 3) Accertati di aver inserito le istruzioni tra parentesi graffe con gli appropriati caratteri di controllo (fai riferimento alla "Piccola guida per l'input dei programmi" che trovi in questa stessa rivista).

Siamo spiacenti di non poter rispondere singolarmente alle richieste di informazioni circa i programmi, prodotti o servizi apparsi su **SUPERCOMMODORE**.



di M. Cristuib Grizzi

Una prima occhiata al Commodore C16

Il nuovo C16 è destinato, almeno sul medio termine, a sostituire il VIC 20.

Rispetto a quest'ultimo presenta infatti innovazioni tali da far apparire ormai obsoleto il fratellino minore.

La confezione con la quale viene fornito il C16 comprende console, trasformatore/alimentatore, cavetto di collegamento con il televisore e manuale d'uso in italiano.

La console si presenta con uno chassis identico a quello del VIC 20/C64, ma di colore nero opaco con tasti grigi. La meccanica della tastiera è quella a cui siamo abituati nei computer Commodore, con un ottimo tocco dei tasti, che consente di usare facilmente il computer anche per la stesura ed il trattamento di lunghi testi. Anche qui è presente al completo il set di caratteri grafici Commodore, con questi ultimi serigrafati sulla faccia anteriore dei tasti. In modo maiuscolo - quello standard all'accensione del computer - il carattere grafico serigrafato sulla destra del tasto è ottenibile con la pressione di quest'ultimo insieme allo Shift, mentre quello a sinistra si ottiene con la pressione contemporanea del tasto desiderato insieme al tasto con il logo Commodore. Passando invece al set minuscolo, premendo contemporaneamente Shift e tasto Commodore diventa ottenibile solo il carattere grafico serigrafato sulla sinistra dei tasti, mentre premendo un tasto insieme allo Shift si otterrà il carattere maiuscolo.

I tasti di controllo del cursore sono stati raddoppiati e spostati: ora è disponibile un singolo tasto per ogni direzione di spostamento del cursore (non occorre più premere anche lo Shift per spostarsi a sinistra od in alto), ma la nuova posizione dei tasti crea per i primi tempi qualche sconcerto in chi è abituato ad usare il VIC od il C64: i tasti cursore ora occupano infatti le posizioni prece-



dentemente occupate da +, -, £ e CLR/HOME. Malgrado i tasti cursore siano raddoppiati, il numero globale di tasti è rimasto invariato e si sono verificati molti spostamenti. Ora è disponibile anche un tasto ESC al posto del Backspace ed un tasto HELP al posto di F7. Il tasto RESTORE è scomparso, facendo posto ad un CLR/HOME delle stesse dimensioni. È comunque facile abituarsi rapidamente alla nuova posizione dei tasti sul C16.

Un'altra novità è un tasto di reset (cold start) del computer, posto sul fianco destro della console a lato dell'interruttore di accensione.

Nella parte posteriore del computer è scomparsa la User Port e sono invece presenti una porta per cartucce, l'uscita TV, l'uscita video per monitor, la porta seriale per il drive o stampante ed infine la porta di interfaccia per il registratore. A proposito di quest'ultima c'è da nota-

re che essa è stata modificata rispetto a quella presente sul VIC/C64, obbligando l'utente all'acquisto del registratore dedicato al C16 (per il resto praticamente identico al solito Datassette). Stesso discorso per le due porte Joystick che ora obbligano all'acquisto di prodotti Commodore. Stupisce che una casa ritenuta seria come la Commodore faccia ricorso a questi espedienti per vendere qualche registratore e qualche joystick in più...

Il cavetto di alimentazione, anch'esso con connettore diverso da quello del VIC/C64, si collega al solito posto sul fianco destro della macchina.

Le funzioni dei nuovi tasti.

ESC: non è, come il nome potrebbe far pensare, un tasto di reset (warm start) del computer, ma permette di accedere ad un certo numero di funzioni di editing dello schermo: premendolo contempo-

OPEN



Una prima occhiata al Commodore C16

raneamente ad ugo tra 18 tasti diversi si accede a varie funzioni di gestione finestre video, scrolling dello schermo, pulizia di quest'ultimo, ecc.

FLASH: premendo CTRL '+', si accede alla funzione di FLASH ON, che provoca il lampeggio di tutto ciò che viene successivamente digitato con la frequenza del cursore. Premendo CTRL '+.' si accede a FLASH OFF che disattiva il lampeggio.

HELP: premendo questo tasto dopo aver ricevuto un messaggio di errore in un programma BASIC, verrà listata la linea in cui si è verificato l'errore e lampeggerà la parte di linea a partire dalla quale si è verificato l'errore. Purtroppo questo tasto ha scarsa utilità nel caso si riceva uno dei messaggi d'errore più diffusi: il tragico "Illegal quantity error in XXXX", tipico di linee che leggono istruzioni DATA e quindi operano una POKE del valore letto, in una certa locazione di memoria. In un caso come questo, premendo HELP lampeggerà la parte di linea contenente la POKE, non fornendo così alcun aiuto su dove cercare il valore errato nelle istruzioni DATA.

TASTI FUNZIONE: mentre nel VIC/C64 questi tasti hanno semplicemente associato un codice ASCII, nel C16 tengono fede alla loro denominazione, in quanto ad ognuno di essi è realmente associata una funzione, e cioè un'istru-

zione BASIC. Le funzioni associate a questi tasti sono GRAPHIC (accede ad una pagina grafica), DLOAD (carica da disco), DIRECTORY (lista su video la directory del disco senza perdita del programma in memoria), SCNCLR (cancella lo schermo), DSAVE (salva su disco), RUN e LIST. Come già detto, l'ultimo tasto funzione corrisponde ad HELP. Tutti i tasti funzione sono ridefinibili da BASIC usando l'istruzione KEY.

RESET: la pressione del tasto di reset sul fianco destro del computer provoca una chiamata alla routine di cold start del Sistema Operativo del C16, riportandolo nelle condizioni del momento di accensione.

II BASIC

Il BASIC è il punto forte del C16. Un gran numero di nuove istruzioni sono state implementate rispetto al BASIC del VIC/C64, permettendo una completa gestione della grafica in alta risoluzione (GRAPHIC, DRAW, CHAR, BOX, CIRCLE, PAINT, LOCATE, SCALE, SSHAPE, GSHAPE), del suono (VOL, SOUND), dell'unità a dischi (BACKUP, COLLECT, COPY, DIRECTORY, DLOAD, DSAVE, HEADER, RENAME, SCRATCH). Sono finalmente disponibili l'IF-THEN-ELSE ed il DO-LOOP-WHILE-UNTIL-EXIT, l'AUTO-

(che genera numeri di linea), il RENUMBER, il DELETE, il COLOR (colore fondo, bordo, testo, multicolore e grado di luminosità), la GETKEY (aspetta che venga premuto un tasto qualunque), la JOY (controllo del joystick), la PRINT USING, le RESUME e TRAP (per la gestione degli errori), e le TRON e TROFF (per il Trace su video).

Una nota di plauso è meritata dall'istruzione MONITOR, che attiva appunto un Monitor (simile al SUPERMONITOR da noi pubblicato per il C64), con tutte le possibilità che questo apre a chi programmi in linguaggio macchina.

Il C16 non dispone tuttavia di SPRITE e delle notevoli possibilità del chip sonoro SID implementato sul C64, ma, con istruzioni grafiche e sonore così potenti, non sfigura certamente al suo fianco.

Lo spazio disponibile per programmi BASIC ammonta a 12277 byte, più che sufficienti per programmi di media lunghezza, e lo schermo di 25 righe per 40 colonne permette una buona visualizzazione dei testi e dei grafici (la risoluzione grafica di 200 per 320 pixel è uguale a quella del C64).

Per concludere, la solita nota di demerito sul manuale fornito in dotazione che, oltre a non fornire nemmeno a grandi linee una mappa di memoria della macchina, non fa cenno - e questo è grave - su quali siano le istruzioni implementate nel Monitor incorporato.

Per ricevere le riviste arretrate di SUPERCOMMODORE con cassetta compilate e spedite il coupon qui sotto!

SUPERCOMMODORE

CEDOLA DI ORDINAZIONE RIVISTE ARRETRATE CON CASSETTA ALLEGATA da compilare e spedire in busta chiusa a
J.soft - Viale Restelli, 5 - 20124 Milano - Tel. 6889228

Vogliete inviarmi i numeri

di SUPERCOMMODORE con cassetta Anno

Al prezzo di L. 15.000

Contanti allegati Assegno allegato n°

Ho spedito l'importo a mezzo vaglia postale

Ho versato l'importo sul cc. n° 19445204 intestato a J. soft - Milano

Pagherò in contrassegno al postino al ricevimento

BUONO D'ORDINE PER RIVISTE ARRETRATE

Anche se l'ordine riguardasse la sola cassetta questa verrà comunque inviata insieme alla rivista al prezzo indicato.

Nome

Cognome

Via

Città C.A.P. Prov.

Se richiesta fattura:

Cod. F. e P. Iva

Data

Firma

Per i residenti all'estero - pagamento anticipato (vaglia o versamento su ns. ccp)



CERCA:

ambosessi di qualsiasi età, residenti in Italia
o all'estero

REQUISITI RICHIESTI:

forte interesse per gli home computer Commodore
disponibilità tempo libero per appassionante
lettura di "superivista" dedicata

OFFRE:

abbonamento a 11 numeri di SuperCommodore
al **prezzo speciale di**
L. 66.000
compresa cassetta
oppure
L. 30.000
per la sola rivista

inviando il coupon pubblicato a fondo pagina otterrete pronta soddisfazione alla vostra richiesta.

Abbonarsi è semplice! Effettuate il versamento con l'apposito modulo c.c.p. inserito in questo fascicolo, oppure ritagliate il tagliando abbonamenti pubblicato in questa pagina e spedite lo allegando un assegno intestato a: J. soft - Viale Restelli, 5 - 20124 Milano.

Tagliando abbonamento a **SUPECOMMODORE** da inviare in busta chiusa a:
J.soft - Viale Restelli 5 - 20124 Milano

- Abbonamento a 11 numeri di **SUPERCOMMODORE** al prezzo speciale di L. 30.000
- Abbonamento a 11 numeri di **SUPERCOMMODORE** + cassetta con tutti i programmi pubblicati al prezzo speciale di L. 66.000

cognome

nome

via

città

cap.

provincia

data

firma

...then



Come sfruttare la tecnica di memorizzazione dei programmi nel VIC 20 e nel C64

di A. Borra

L'idea alla base di questo articolo deriva dal metodo di memorizzazione dei programmi realizzati nel VIC 20 e nel C64.

Vedremo in particolare come sfruttare alcune caratteristiche per ottenere una prima routine che permetta di evidenziare i commenti (rendendo più leggibili i listati) ed una seconda che abbia lo scopo di proteggere un qualsiasi programma residente in memoria (protezione "statica", come vedremo).

Entrambe le routine (realizzate per il VIC 20) funzionano anche sul C64.

Memorizzazione di un programma

Un programma BASIC viene memorizzato come una successione di blocchi (o record) di lunghezza variabile a partire dall'inizio della memoria. Ogni record rappresenta una linea di programma ed ha la seguente struttura:

1°-2° byte: indirizzo di link (1° byte+2° byte*256)

3°-4° byte: numero della linea BASIC (3° byte+4° byte*256)

5°-(n-1)° byte: istruzioni
n° byte: 0 (=separatore di record)

L'indirizzo di link permette il concatenamento di due linee, andando a puntare all'indirizzo di link della istruzione successiva.

Poiché l'ultimo record è seguito da due byte contenenti zero, l'ultima linea codificata punta a tali due byte.

Per capire bene il funzionamento del link, provate a digitare questo semplice programmino (senza modificare nulla):

```
10 A=5
20 PRINT "A VALE";
30 PRINTA
```

provatelo, listatelo e quindi battete:

```
A = PEEK (43) + PEEK (44) * 256 :
POKEA, PEEK (A) + 15 : LIST +(R)
```

e vedrete che la linea 20 non compare più, nonostante ci sia ancora (e basta dare il RUN per sincerarsene). In effetti abbiamo modificato l'indirizzo di link della prima istruzione, facendo in modo che puntasse alla terza e non più alla seconda e così la routine di LIST non ha "visto" la seconda istruzione.

La routine di RUN, invece, ha conside-

rato le istruzioni così come erano in memoria senza considerare il link.

Consideriamo ora la codifica della etichetta della linea BASIC: in teoria si potrebbero avere etichette tra 0 e 65535, però in pratica il massimo numero di linea ammesso è 63999. Quindi nessuna istruzione di etichetta maggiore può essere immessa (ma nemmeno cancellata!) in modo diretto. Nulla ci vieta, però, di modificare l'etichetta di una linea di un programma già residente in memoria mediante delle POKE. Tornando al programmino considerato, provate a battere:

```
A = PEEK (43) + PEEK (44) * 256 :
POKEA + 2,255 : LIST +(R)
```

e osserverete che ora la prima istruzione ha 255 come etichetta.

La cosa però non è molto "pulita": provate infatti a battere:

```
40 END +(R)
```

e a listare: tale riga è diventata la prima, nonostante la presenza di ben due righe di etichetta minore! Ma la cosa più grave è un'altra: provate ad esempio a digitare:

```
GOTO 20 +(R)
```



vedrete comparire il messaggio "UN-DEF'D STATEMENT", nonostante la riga 20 sia ben presente! Il problema è che il BASIC del VIC (e del C64) va a considerare le etichette delle righe presenti in memoria partendo dalla prima che trova: in entrambi i casi la presenza dell'etichetta 255 lo fa agire in modo apparentemente non corretto.

Quindi bisogna fare molta attenzione nel modificare un programma già residente in memoria tramite le POKE.

E veniamo infine alla codifica delle istruzioni. Come si vede dalla tabella, ogni carattere ha un suo codice ed inoltre le parole riservate del BASIC sono codificate con un opportuno valore, in modo da ridurre l'occupazione di memoria dei programmi. Sempre dalla tabella si osserva che alcuni codici (1-31 e 204) non corrispondono né ad una parola BASIC né ad un carattere. Alcuni di tali valori hanno però degli effetti particolari.

A questo punto abbiamo tutte le conoscenze necessarie per affrontare le due routine cui ho accennato, le quali, come si sarà capito, prendono spunto dalla possibilità di modificare un programma già residente in memoria agendo direttamente sui record rappresentanti le linee del programma.

Come evidenziare i commenti

I commenti (ottenuti mediante l'istruzione REM) sono una parte molto utile in un programma BASIC, in quanto spiegano il flusso di controllo ed aiutano la ricerca di particolari routine.

Certo che se tali commenti fossero "videati" in reverse o in un colore diverso dal colore delle istruzioni (esempio in rosso o in verde), allora sarebbe ancora più utile il loro utilizzo, diventando immediata la loro individuazione. Si dà il caso che proprio alcuni dei codici non utilizzati dalla routine di codifica dei programmi permettano di cambiare colore o di videare in reverse. Tali codici sono:

- 5 scrive in bianco
- 28 scrive in rosso
- 30 scrive in verde
- 31 scrive in blu
- 18 scrive in reverse

I primi quattro codici sono globali, nel senso che modificano il colore anche delle linee seguenti quella in cui compaiono, mentre il codice di reverse è locale, cioè la riga seguente viene mostrata nei caratteri normali.

Tali codici non vengono riconosciuti dalla routine di decodifica delle linee BASIC e quindi possono essere posti solo dopo una istruzione di REM (codice 143). E ciò ci basta.

Se, infatti, dopo aver digitato un programma, andiamo ad inserire dopo ogni codice 143 il codice 28 e poi inseriamo il codice 31 prima dello 0, che indica la terminazione della riga, otteniamo tutti i commenti di colore rosso, senza però

TAVOLA DI CODIFICA DEI CARATTERI				
0	FINE LINEA	64 @	128 END	192 TAN
1	NON USATO	65 A	129 FOR	193 ATH
2	NON USATO	66 B	130 NEXT	194 PEK
3	NON USATO	67 C	131 DATA	195 LEN
4	NON USATO	68 D	132 INPUT#	196 STR#
5	NON USATO	69 E	133 INPUT	197 VAL
6	NON USATO	70 F	134 DIM	198 ASC
7	NON USATO	71 G	135 READ	199 CHR#
8	NON USATO	72 H	136 LET	200 LEFT#
9	NON USATO	73 I	137 GOTO	201 RIGHT#
10	NON USATO	74 J	138 RUN	202 MID#
11	NON USATO	75 K	139 IF	203 GO
12	NON USATO	76 L	140 RESTORE	204 NON USATO
13	NON USATO	77 M	141 GOSUB	205 FOR
14	NON USATO	78 N	142 RETURN	206 NEXT
15	NON USATO	79 O	143 REM	207 DATA
16	NON USATO	80 P	144 STOP	208 INPUT#
17	NON USATO	81 Q	145 ON	209 INPUT
18	NON USATO	82 R	146 WAIT	210 DIM
19	NON USATO	83 S	147 LOAD	211 READ
20	NON USATO	84 T	148 SAVE	212 LET
21	NON USATO	85 U	149 VERIFY	213 GOTO
22	NON USATO	86 V	150 DEF	214 RUN
23	NON USATO	87 W	151 POKE	215 IF
24	NON USATO	88 X	152 PRINT#	216 RESTORE
25	NON USATO	89 Y	153 PRINT	217 GOSUB
26	NON USATO	90 Z	154 CONT	218 RETURN
27	NON USATO	91 [155 LIST	219 REM
28	NON USATO	92 £	156 CLR	220 STOP
29	NON USATO	93 J	157 CMD	221 ON
30	NON USATO	94 †	158 SYS	222 WAIT
31	NON USATO	95 +	159 OPEN	223 LOAD
32	(SPAZIO)	96 -	160 CLOSE	224 SAVE
33	!	97 ^	161 GET	225 VERIFY
34	"	98	162 NEW	226 DEF
35	#	99 -	163 TAB<	227 POKE
36	*	100 -	164 TO	228 PRINT#
37	%	101 -	165 FN	229 PRINT
38	&	102 -	166 SPC<	230 CONT
39	/	103	167 THEN	231 LIST
40	<	104	168 NOT	232 CLR
41	>	105 \	169 STEP	233 CHD
42	*	106 \	170 +	234 SYS
43	+	107 /	171 -	235 OPEN
44	.	108 L	172 *	236 CLOSE
45	-	109 \	173 /	237 GET
46	.	110 /	174 †	238 NEW
47	/	111 ¶	175 AND	239 TAB<
48	0	112 ¶	176 OR	240 TO
49	1	113 •	177 >	241 FN
50	2	114 -	178 =	242 SPC<
51	3	115 •	179 <	243 THEN
52	4	116	180 SH	244 NOT
53	5	117 /	181 INT	245 STEP
54	6	118 x	182 ABS	246 +
55	7	119 0	183 USR	247 -
56	8	120 *	184 FRE	248 *
57	9	121	185 POS	249 /
58	.	122 •	186 SQR	250 †
59	;	123 +	187 RND	251 AND
60	<	124 ¶	188 LOG	252 OR
61	=	125	189 EXP	253 >
62	>	126 ¶	190 COS	254 =
63	?	127 ¶	191 SIN	255 ¶

giudicare l'esecuzione del programma. Naturalmente, nel fare ciò bisogna stare attenti che il fatidico codice 143 non capiti nei 4 byte indicanti il link o il numero di etichetta della linea BASIC, pena risultati catastrofici. In pratica, l'idea è questa: data per esempio la linea

10 REMPIPO

che viene codificata:

LB LA 10 0 143 80 73 80 80 79 0

(LB=link basso, LA=link alto) si vorrebbe ottenere: ,

LB' LA' 10 0 143 28 80 73 80 80 79 31 0

In questo modo, però, bisogna spostare di posto in memoria i vari record e quindi vanno aggiornati gli indirizzi di link di ogni riga del programma. È invece più



Come sfruttare la tecnica di memorizzazione dei programmi nel VIC 20 e nel C64

facile scrivere i codici di controllo "sopra" il primo e l'ultimo carattere del commento, in modo da ottenere:

```
LB LA 10 0 143 28 73 80 80 31 0
```

così, sul video, si leggerebbe solo

```
10 REMIPP
```

con "IPP" scritto in rosso. Ma questo non crea grossi problemi: è sufficiente porre un carattere qualsiasi (per esempio l'asterisco) prima e dopo il commento:

```
10 REM-PIPPO-
```

Il programma riportato in fondo all'articolo permette appunto di evidenziare i commenti di un qualsiasi programma residente in memoria, operando proprio come descritto. Il programma BASIC crea una zona protetta alla fine della memoria: 77 byte nei quali carica una routine in linguaggio macchina (che è quella che modifica i commenti). Alla fine il programma BASIC segnala a quale indirizzo parte la routine (attivabile mediante una SYS). Essendo il programma BASIC e la routine scritte in forma parametrica, funzionano sia sul VIC con qualunque tipo di espansione che sul C64. Il programma BASIC dopo la sua esecuzione non serve più: si può digitare NEW e caricare o digitare un nuovo programma.

La routine si può attivare in qualsiasi momento, poiché rimane in memoria in zona protetta fino allo spegnimento della macchina.

Si può notare che:

- la routine termina se non c'è nessun programma in memoria o dopo aver considerato tutte le righe del programma;

- se un record non contiene il codice 143 (=REM) tra i codici delle istruzioni, si considera il record successivo;

- una linea BASIC può contenere anche altre istruzioni oltre al comando REM (esempio 100 PRINTA: REM*STAMPA*);

- se un REM non ha commento (esempio 100 REM), non viene modificato nulla e si passa al record successivo;

- in caso di REM con commento vengono modificati il primo carattere dopo il REM e l'ultimo carattere prima dello 0;

- se un REM ha un commento di un unico carattere (esempio 100 REM*), questo viene modificato due volte;

- la routine può essere chiamata più volte, in quanto "mangia" due caratteri solo la prima volta che opera su un record.

Dunque, la routine effettua molti controlli, poiché un suo eventuale errore potrebbe pregiudicare in modo irreparabile il programma residente in memoria. Consigliamo quindi di prestare molta attenzione nella battitura del programma BASIC di caricamento e di salvarlo prima di eseguire la routine stessa.

Per chi ha il C64 c'è il problema che il colore "standard" (cioè quello che si ha all'accensione della macchina) delle scritte è l'azzurro, e questo non è un colore ottenibile con il metodo appena descritto. Quindi bisogna accontentarsi di ottenere i commenti solo in reverse, poiché se si cambia il colore dei commenti, non si riesce a tornare alla scrittura in azzurro del resto del listato. Va quindi modificata la linea 1130 del programma (quella contenente il codice di ritorno alla scrittura "standard", che è 31 per il VIC) in questo modo:

```
1130 DATA 042
```

che mette un asterisco come ultimo carattere del commento.

Nel caso ci si dimentichi l'indirizzo di partenza della routine, questo si può riottenere mediante:

```
PRINT PEEK(55)+PEEK(56)*256+(R)
```

Protezione "statica" di un programma

I due articoli di Alessandro Guida apparsi in Personal Software n° 5 e n° 14 spiegano molto bene come disabilitare il tasto di RUN/STOP e come evitare il LIST di un programma. L'unico difetto è che la protezione dal LIST si può attuare solo durante l'esecuzione del programma e non prima: se si accende la macchina e si carica un programma, si può sempre listararlo prima della sua esecuzione.

In questo senso le protezioni descritte possono intendersi come dinamiche, in quanto hanno luogo solo dopo aver dato il RUN al programma. Se invece si sfrutta opportunamente il modo con cui viene memorizzato un programma, è possibile ottenere una protezione "statica" dal listing, nel senso che il programma non è mai listabile prima della sua esecuzione.

Vedremo però che il programma sarà listabile DOPO l'esecuzione, ma a questo punto gli utili "trucchi" di Guida permettono di completare l'opera, rendendo più sicuro il proprio software.

L'idea è sempre quella di utilizzare qualcuno di quei codici che non vengono utilizzati dalla routine di codifica di una linea BASIC. In particolare 2 codici

fanno al caso nostro:

20: corrisponde all'azione del tasto INST/DEL

204: stampa "SYNTAX ERROR" e termina il listing

Si può inoltre sfruttare il fatto che una linea non può essere né inserita né cancellata, se la sua etichetta è maggiore di 63999.

D'altra parte non è possibile avere come prima istruzione una linea con etichetta minore di 64000, altrimenti la protezione dal LIST non sarebbe perfetta: l'idea è allora quella di modificare tale etichetta durante l'esecuzione. È sufficiente che la seconda linea del programma si occupi di ciò. Dunque, per riassumere, un programma per essere protetto in modo "statico" dal LIST deve avere:

- come prima riga: istruzione di etichetta maggiore di 63999, con una istruzione di REM seguita da un numero sufficiente di codici 20, tali da cancellare dal video le scritte della linea, e da un codice 204 per stampare "SYNTAX ERROR";

- come seconda riga: istruzione che cambia l'etichetta della prima riga.

La cosa sarebbe però noiosa, se fatta per ogni programma: da qui l'idea di una routine "protettiva", che aggiunga tali linee ad un qualsiasi programma residente in memoria. Tale routine (che viene caricata, analogamente alla routine precedente, da un programma BASIC in zona protetta, cioè gli ultimi 160 byte della memoria utente) è realizzata in linguaggio macchina e in effetti non aggiunge alcuna linea: si limita a modificare la prima linea (che deve avere un certo formato) "spezzandola" in due linee. Vediamo come funziona la routine:

- la prima linea del programma da proteggere deve avere una prima istruzione costituita da una REM seguita da ALMENO 53 caratteri (esempio 53 asterischi). Altrimenti la routine non compie alcuna modifica;

- il link di tale linea punta al secondo record presente in memoria. Poiché la routine spezza tale primo record in due, l'indirizzo deve essere salvato per poi essere messo come link del record "1 bis" (creato dalla routine);

- bisogna calcolare il link del primo (nuovo) record. Tale calcolo è eseguito con estrema accuratezza per evitare errori;

- la routine scrive la prima linea, mettendogli 65535 come etichetta;

- la routine scrive la seconda linea:

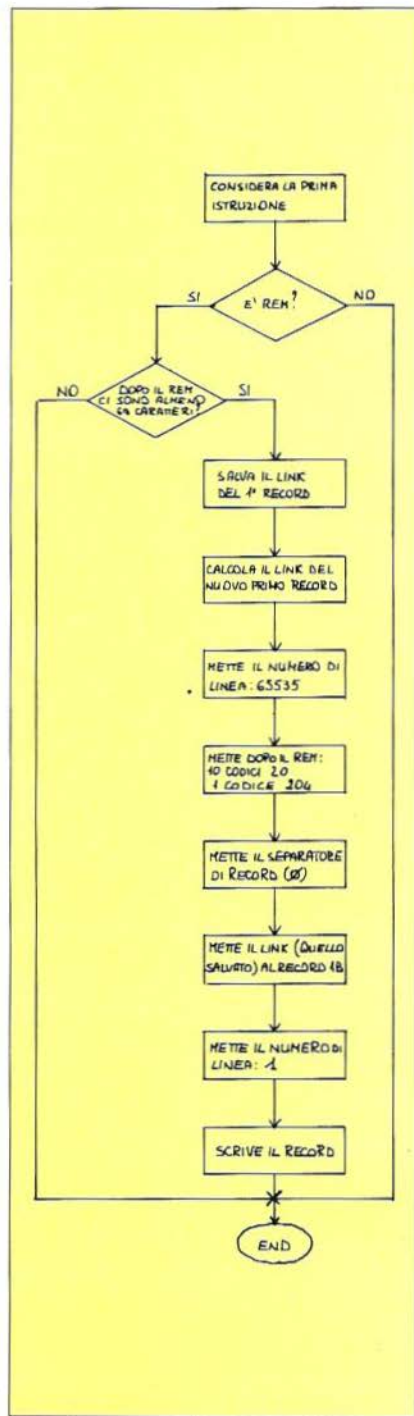
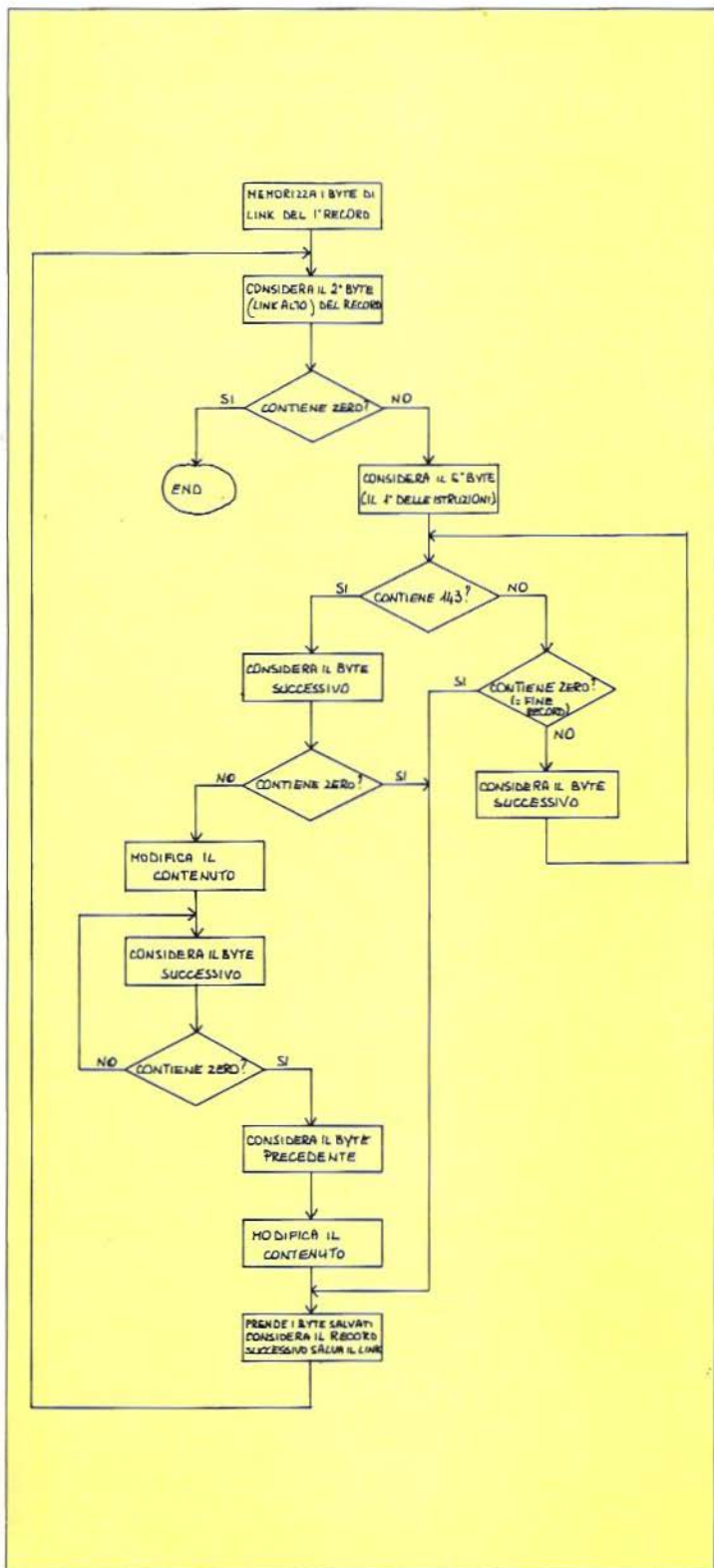


fig. 1 - Diagramma di flusso della 1ª routine

fig. 2 - Diagramma di flusso della 2ª routine



Come sfruttare la tecnica di memorizzazione dei programmi nel VIC 20 e nel C64

A= PEEK (43) + PEEK (44) * 256 :
POKEA + 2,0 : POKEA + 3,0 : A = 0:
REM

mettendogli 1 come etichetta. L'azione di questa riga è quella di modificare, mettendola a 0, l'etichetta della prima riga, in modo da permettere (durante l'esecuzione del programma, ma non prima!) i GOTO. La variabile A viene azzerata nel caso sia utilizzata nel programma da proteggere. La REM viene posto nel caso la riga originale del programma (quella contenente almeno 53 caratteri dopo la REM, per intenderci) avesse contenuto più di 53 caratteri. Quindi, per proteggere dal LIST un programma le azioni da compiere sono:
- caricare in memoria la routine;
- caricare il programma da proteggere;

- verificare che tale programma non contenga istruzioni con etichetta 0 oppure 1;
- aggiungere al programma una linea del tipo 0 REM seguita da 53 asterischi (almeno);
- attivare la routine in linguaggio macchina mediante la SYS opportuna (esempio: SYS PEEK(55)+PEEK(56)*256);
- verificare la protezione e quindi salvare (PRIMA di eseguire!!).
Se qualcuno per curiosità volesse vedere la nuova riga creata dalla routine, può eseguire un qualsiasi programma protetto e quindi battere 0 (per cancellare la riga che impedisce il listing) e digitare

LIST 1 + (R)
Sia il programma BASIC che la routine



funzionano senza modifiche sia sul VIC 20 (con qualunque tipo di espansione di memoria) che sul C64.

Protettore di programmi

```
0 REM*****:rem 0
1 REM** **:rem 189
2 REM** **:rem 190
3 REM** PROTETTORE DI PROGRAMMI **:rem 18
4 REM** **:rem 192
5 REM** **:rem 193
6 REM** A. BORRA - 1984 **:rem 170
7 REM** **:rem 195
8 REM** **:rem 196
9 REM*****:rem 9
100 PRINT TAB(255) TAB(182) "ATTENDERE" T
AB(255) :rem 73
109 REM** CALCOLA FINE MEMORIA **:rem 65
110 FM = PEEK(55)+256*PEEK(56) :rem 21
120 A = INT ((FM-160)/256) : B = (FM-160)
-A*256 :rem 87
129 REM** MODIFICA INIZIO VARIABILI **:rem 201
130 POKE 51,B : POKE 52,A :rem 212
139 REM** MODIFICA FINE MEMORIA **:rem 145
140 POKE 55,B : POKE 56,A :rem 221
150 NF = FM-160 :rem 190
159 REM** CARICA ISTRUZIONI L.M. IN ZONA
PROTETTA **:rem 37
160 FOR I = 0 TO 159 :rem 118
```

```
170 READ V :rem 10
180 POKE NF+I,V :rem 82
190 NEXT :rem 217
200 PRINT TAB(202) "PER ESEGUIRE:" : PRIN
T TAB(27) "SYS";NF TAB(200) :rem 189
210 END :rem 106
999 REM** ISTRUZIONI IN L.M. **:rem 227
1000 DATA 160,004,177,043,201,143,240,001
,096,162 :rem 82
1010 DATA 000,200,177,043,201,000,208,001
,096,232 :rem 68
1020 DATA 224,053,208,243,160,001,177,043
,133,252 :rem 90
1030 DATA 136,177,043,133,251,162,000,165
,043,024 :rem 91
1040 DATA 105,017,144,001,232,145,043,165
,044,224 :rem 87
1050 DATA 000,240,003,024,105,001,200,145
,043,200 :rem 55
1060 DATA 169,255,145,043,200,145,043,200
,162,000 :rem 88
1070 DATA 169,020,200,145,043,232,224,010
,208,248 :rem 91
1080 DATA 200,169,204,145,043,200,169,000
,145,043 :rem 91
1090 DATA 200,165,251,145,043,200,165,252
,145,043 :rem 95
1100 DATA 162,121,134,253,200,132,254,164
,253,177 :rem 94
1110 DATA 055,164,254,145,043,232,224,160
,208,238 :rem 103
1120 DATA 096,001,000,065,178,194,040,052
,051,041 :rem 89
1130 DATA 170,194,040,052,052,041,172,050
,053,054 :rem 88
1140 DATA 058,151,065,170,050,044,048,058
,151,065 :rem 107
1150 DATA 170,051,044,048,058,065,178,048
,058,143 :rem 120
```

**Programma
evidenziatore di RcM**

```

0 REM*****
1 REM**                                     :rem 0
  **                                     :rem 189
2 REM**                                     :rem 190
  **                                     :rem 9
3 REM**  EVIDENZIATORE DI REM             :rem 192
  **                                     :rem 193
4 REM**                                     :rem 170
  **                                     :rem 195
5 REM**                                     :rem 196
  **                                     :rem 9
6 REM**  A. BORRA - 1984                   :rem 73
  **                                     :rem 65
7 REM**                                     :rem 21
  **                                     :rem 5
8 REM**  MODIFICA INIZIO VARIABILI **     :rem 201
  **                                     :rem 212
9 REM**                                     :rem 145
100 PRINT TAB(255) TAB(182) "ATTENDERE" T :rem 221
    AB(255)                               :rem 149
109 REM** CALCOLA FINE MEMORIA **
110 FM = PEEK(55)+256*PEEK(56)
120 A = INT ((FM-77)/256) : B = (FM-77)-A
    *256
129 REM** MODIFICA INIZIO VARIABILI **
130 POKE 51,B : POKE 52,A
139 REM** MODIFICA FINE MEMORIA **
140 POKE 55,B : POKE 56,A
150 NF = FM-77
  
```

```

159 REM** CARICA ISTRUZIONI L.M. IN ZONA
    PROTETTA ** :rem 37
160 FOR I = 0 TO 76 :rem 68
170 READ V :rem ;10
180 POKE NF+I,V :rem 82
190 NEXT :rem 217
200 PRINT TAB(202) "PER ESEGUIRE:" : PRIN
    T TAB(27) "SYS";NF TAB(200) :rem 189
210 END :rem 106
999 REM** ISTRUZIONI IN L.M. ** :rem 227
1000 DATA 165,043,133,251,165 :rem 137
1010 DATA 044,133,252,160,001 :rem 124
1020 DATA 177,251,201,000,208 :rem 129
1030 DATA 001,096,160,004,177 :rem 136
1040 DATA 251,201,143,240,009 :rem 129
1050 DATA 201,000,240,032,200 :rem 112
1060 DATA 169,000,240,241,200 :rem 128
1070 DATA 177,251,201,000,240 :rem 130
1080 DATA 020,169 :rem 65
1086 REM** CONTIENE IL CODICE PER IL CAMB
    IAMENTO DELLE SCRITTE ** :rem 12
1087 REM** 005=BIANCO :rem 028=ROSS
    O ** :rem 71
1088 REM** 030=VERDE :rem 031=BLU
    ** :rem 87
1089 REM** 018=REVERSE ON ** :rem 237
1090 DATA 018 :rem 125
1100 DATA 145,251,200,177,251 :rem 135
1110 DATA 201,000,240,004,169 :rem 122
1120 DATA 000,240,245,136,169 :rem 137
1129 REM** CONTIENE IL CODICE PER TORNARE
    ALLA SCRITTURA NORMALE ** :rem 213
1130 DATA 031 :rem 115
1140 DATA 145,251,160,000,177 :rem 136
1150 DATA 251,170,200,177,251 :rem 138
1160 DATA 134,251,133,252,169 :rem 146
1170 DATA 000,240,187 :rem 1
  
```

E' IN EDICOLA

PC

MAGAZINE

*La rivista dei
sistemi MS-DOS*

*La guida completa
del personal
computer IBM
e compatibili*

con tutta la competenza del



GRUPPO EDITORIALE JACKSON

REM:HW

Il Joystick

Come acquisire i segnali provenienti dai joystick e simularne il funzionamento sulla tastiera: seconda parte.

Come simulare il funzionamento di un joystick sulla tastiera del C64.

Vorremmo innanzi tutto precisare alcune delle caratteristiche del programma (Programma 2) che accompagna questa sezione.

Non abbiamo ritenuto conveniente spingere la simmetria tra questo programma ed il precedente fino a simulare "due" joystick sulla tastiera. Tutto ciò che viene realizzato tramite i tasti indicati nel programma può essere replicato senza problemi in qualsiasi altra parte della tastiera tramite una diversa serie di tasti.

Abbiamo preferito invece dimostrare le diversità esistenti tra due possibili modi di acquisizione del segnale da tastiera, vale a dire tramite le istruzioni PEEK (197) e GET X\$. Ulteriori varianti sarebbero possibili per quanto riguarda il meccanismo dei test nel secondo caso (direttamente sulla variabile alfanumerica contenente il carattere acquisito, invece che sulla conversione in codice ASCII del carattere stesso), e per la realizzazione di un pulsante di fuoco indipendente dai due meccanismi di acquisizione suddetti. Entrambe queste opzioni, per ragioni di semplicità, non sono state inserite nel programma, ma verranno brevemente affrontate in questa stessa sezione.

Il primo problema che ci si trova ad affrontare, se si desidera replicare tramite i tasti il funzionamento di un joystick, è rappresentato dal fatto che, mentre gli spostamenti di quest'ultimo sono controllabili tramite solo quattro microinterruttori, una serie di quattro tasti non permette di comprendere gli spostamenti lungo le direzioni intermedie.

Dato infatti che l'acquisizione da tastiera, sia che avvenga sotto forma di istruzione PEEK che di GET, scandisce l'intera tastiera, non è possibile comporre i segnali provenienti da due tasti diffe-

Hardware

di U. G. Barzaghi



renti premuti contemporaneamente, dal momento che entrambi i suddetti meccanismi di acquisizione segnalano la pressione solo del primo dei due (ed una lieve differenza tra l'attivazione dei due tasti esiste sempre).

Il problema è, comunque, meno grave di quel che sembra. È solo necessario introdurre una certa ridondanza rispetto al segnale dei joystick, definendo il doppio di microinterruttori, cioè otto tasti invece che quattro.

Con i due metodi suddetti è necessario escludere solo i tasti RUN/STOP (per ovvie ragioni), CTRL, RESTORE, SHIFT, SHIFT LOCK e COMMODOR-

RE (in quanto inerti, se premuti da soli). Ciò lascia ampie possibilità di scelta (notate come siano compresi i tasti funzione che, in alcuni casi, si rivelano particolarmente comodi), anche per quanto riguarda il posizionamento del nostro joystick "virtuale" (o, addirittura, di più d'uno) sulla tastiera.

Dovendo disporre di tutte le quattro direzioni cardinali e delle quattro intermedie, è possibile, con un po' di fantasia, arrivare a definire fino a tre joystick sulla tastiera, benché il loro uso risulterebbe sicuramente ben poco agevole, ad esempio con la seguente definizione dei tasti:

Joystick 1	Joystick 2	Joystick 3
1 2 3	T Y U	O + -
Q W E	G H J	O P @
A S D	B N M	L ; :

Le varie direzioni sono stabilite dalla posizione relativa tra i tasti periferici ed il tasto centrale di ciascun joystick (rispettivamente W, H e P).

Ciò significa che, ad esempio, premendo 'S' l'utente del primo joystick virtuale provocherebbe uno spostamento verso il basso; premendo il tasto 'T' il secondo giocatore si sposterebbe verso l'alto e a sinistra e premendo @ il terzo giocatore si muoverebbe verso destra.

Per quanto riguarda i pulsanti di fuoco, potrebbe essere una buona idea utilizzare i tasti centrali di ciascun gruppo (altrimenti inutilizzati).

Il caso suddetto è, naturalmente, un po' al limite; riteniamo, comunque che in caso di necessità l'implementazione di un paio di joystick "virtuali", "strategicamente" disposti, non provochi particolari problemi.

Una caratteristica da tenere ben presente, comunque, è rappresentata dal fatto che, siccome l'esame delle varie direzioni deve essere esaustivo, la decodifica di più "joystick" provoca un rallentamento del programma.

Altri aspetti caratteristici vengono determinati dal diverso sistema di acquisizione del segnale premuto.

Tramite l'istruzione PEEK (197) si legge una locazione di pagina zero del calcolatore, contenente un valore numerico compreso tra 0 e 255. Dal momento che non esistono tavole dei valori corrispondenti ai tasti letti in questo modo, abbiamo ritenuto opportuno inserire nel programma una breve routine, che vi permetterà di stabilire il valore associato a ciascun tasto, ad esempio se desiderate ridefinire nel programma i tasti associati alle varie direzioni.

Per attivare questo sottoprogramma è sufficiente, dopo aver copiato il Programma 2, battere in formato diretto RUN950. Nell'angolo superiore dello schermo comparirà il valore numerico 64, che corrisponde al caso in cui la tastiera sia inattiva, cioè non sia premuto alcun tasto in quel momento; è quindi possibile compiere semplici test in questo modo per stabilire se il nostro "joystick" viene o no utilizzato.

Tenendo premuto uno dei tasti, con esclusione di quelli suddetti, si vedrà comparire la cifra corrispondente, acquisita tramite l'istruzione PEEK(197). Attenzione al fatto che l'acquisizione con questo sistema, contrariamente all'istruzione GET, come vedremo, non dispone di un'area tampone in cui memorizzare una certa quantità dei tasti premuti in successione; le cifre che compariranno sullo schermo, pertanto, corrisponderanno ai tasti "tenuti" premuti; non appena essi verranno rilasciati ricomparirà il valore numerico corrispondente a "tastiera inerte".

Per avere una rapida dimostrazione battete

FORI = 1TO10000:?[CLR] PEEK (197): NEXT

Gli stessi valori si possono ottenere leggendo la locazione 203 di pagina zero in formato analogo.

L'acquisizione tramite istruzione GET

Il meccanismo di acquisizione che utilizza l'istruzione GET è, invece, sostanzialmente differente. Come detto, esso è, innanzi tutto, fornito di buffer; è pertanto possibile acquisire una serie di dieci tasti in sequenza e riprodurne gli effetti in uscita. Ciò significa che l'area tampone utilizzata dalla tastiera è organizzata come una coda (del tipo che viene generalmente chiamato FIFO, cioè "First In, First Out"; primo in entrata, primo in uscita) avente solo dieci "posti" disponibili; l'acquisizione avviene sempre dal primo carattere della "coda", ed ogni acquisizione fa scorrere in avanti di una posizione tutti i caratteri seguenti.



Questa caratteristica non è sempre desiderabile ed è quindi talvolta necessario preoccuparsi che il buffer della tastiera sia vuoto prima di acquisire ciascun segnale. Un modo per farlo consiste semplicemente nel leggere dieci caratteri tramite un ciclo

FORI=1TO10:GETA\$:NEXT

Un modo più elegante di ottenere lo stesso risultato consiste nell'alterare il contenuto della locazione di pagina zero (198), indicante il numero di caratteri contenuto nel buffer di tastiera. Una istruzione

POKE 198,0

posta dopo ciascuna acquisizione permetterà di ingannare il calcolatore, facendogli trascurare i caratteri già inseriti nell'area tampone.

Una delle cose cui si deve fare attenzione nella gestione di un simile sistema di acquisizione dei segnali è il caso in cui non sia stato premuto alcun tasto. Un

simile caso può essere rilevato con una istruzione del tipo

GET X\$:IFX\$=""THEN...

È importante, però, stabilire che cosa il programma deve fare in questo caso: se non vi sono problemi di tempo o si suppone che gli ingressi siano sufficientemente frequenti, sarà opportuno riciclare sulla stessa istruzione GET in attesa del primo segnale in ingresso; se, invece, i segnali giungono con frequenza relativamente bassa, e non è opportuno che il programma rimanga bloccato in attesa del primo tasto premuto, sarà opportuno diramare verso altre parti o richiamare particolari sottoprogrammi (ad esempio per aggiornare un cronometro o la posizione dei giocatori sullo schermo nel caso di un videogioco), per poi ritornare a questa sezione di acquisizione dei dati.

Per quanto riguarda la decodifica del segnale sono possibili due alternative: una consiste nel compiere i test direttamente sul carattere alfanumerico acquisito tramite la variabile X\$, ad esempio

10 GETX\$=""THEN10
20 IFX\$="W"THEN... : REM NORD

Questo sistema, però, non permette di utilizzare quei tasti, come RETURN, che non hanno un carattere corrispondente. Per ovviare a questo inconveniente è preferibile utilizzare la conversione del carattere alfanumerico nel valore numerico in codice ASCII corrispondente, ottenibile tramite un'istruzione

X=ASC(X\$)

di cui l'Appendice F del "Manuale d'uso del C64" riporta una tavola completa. La decodifica del tasto 'W' del breve programma precedente avrebbe quindi la forma

10 GETX\$:IFX\$=""THEN10
20 X=ASC(X\$):IFX=87THEN... : REM NORD

Prima di commentare le varie opzioni del Programma 2 qualche breve cenno su di un'altra locazione di pagina zero che può essere utilizzata per acquisire segnali dalla tastiera.

Le sue caratteristiche la rendono particolarmente sfruttabile per funzioni di pulsante di fuoco, in particolare quando si desidera che quest'ultimo sia indipendente, ad esempio per ragioni di rapidità di acquisizione, dalla decodifica dei segnali di spostamento del joystick "virtuale".

La locazione in questione può essere letta tramite una istruzione

PEEK(653)

in maniera non differente da quanto fatto per la lettura della tastiera tramite le locazioni 197 o 203. Il contenuto del

REM:HW

Il Joystick

la locazione suddetta è uguale a zero, se nessuno dei tasti da essa controllati viene premuto nel momento della lettura. I tasti in questione rappresentano una parte di quelli esclusi dalla lettura per le due locazioni precedenti, vale a dire i tasti SHIFT (e SHIFT LOCK), COMMODORE e CTRL. Il contenuto della locazione 653, nel caso in cui uno dei tasti suddetti venga premuto, restituisce, rispettivamente, i valori numerici 1, 2 e 4.

Volendo utilizzare i tre tasti suddetti come pulsante di fuoco è possibile inserire nei punti opportuni di qualsiasi programma una istruzione del tipo

```
IFPEEK(653) < > 0 THEN... REM PULSANTE DI FUOCO PREMUTO  
oppure
```

```
IFPEEK(653)=0 THEN... REM PULSANTE DI FUOCO INATTIVO
```

Programma 2: funzionamento e commenti al listato

Il Programma 2 presenta alcune lievi, ma sostanziali, differenze rispetto al programma precedente. Innanzi tutto il menu viene ridotto da tre a due voci, dal momento che, come si è detto, abbiamo preferito rinunciare all'implementazione di due joystick "virtuali" sulla tastiera.

Dopo aver mandato in esecuzione il programma, ed aver atteso che esso trasferisca il contenuto delle frasi DATA in memoria per definire la forma degli sprite, comparirà sullo schermo un menu composto da due sole voci, che permetterà di scegliere tra l'acquisizione del segnale tramite istruzione PEEK (197) o GET X\$.

Date le dimensioni del menu, sarebbe stato superfluo, a rigor di logica, definire sia tasti per il passaggio da un'opzione alla precedente che per spostamenti in senso opposto. Abbiamo comunque preferito, anche a scapito di una certa ridondanza, mantenere le stesse caratteristiche del Programma 1. Una delle ragioni per questa scelta è stata rappresentata dal fatto che il controllo di un indice "verticale" delle voci di un menu costituisce un'ottima possibilità di esemplificare l'impiego dei tasti funzione, che fanno in questo caso le veci dei microinterruttori "NORD" e "SUD" del joystick.

Per quanto riguarda l'acquisizione di questi segnali, date le caratteristiche che contraddistinguono i tasti funzione, abbiamo preferito utilizzare l'istruzione PEEK (197). Il tasto di funzione 'fi' riveste le funzioni della posizione "NORD"

del joystick, mentre il tasto 'f7' assume quelle della posizione "SUD". Una volta attivata l'opzione desiderata è possibile passare al quadro principale del programma di sparo premendo uno dei due pulsanti di fuoco "virtuali" definiti sulla tastiera. Abbiamo preferito dare un duplice esempio delle possibilità di impiego di questa tecnica, definendo come tali sia la barra spaziatrice (secondo noi la più comoda, date le dimensioni e la sua posizione) che il tasto RETURN; ma è ovviamente possibile utilizzare qualsiasi altro tasto letto con una delle due tecniche illustrate.

Una volta attivato il quadro della "busola" tutto si svolge, apparentemente, in maniera non molto differente dal Programma 1. Le uniche differenze riguardano, ovviamente, il nostro joystick virtuale, definito dai tasti

Nord-ovest	Nord	Nord-est
	Q W E	
Ovest	A D Est	
	Z X C	
Sud-ovest	Sud	Sud-est

e dai due pulsanti di fuoco (barra spaziatrice e RETURN).

Le quattro caselle nella parte superiore dello schermo contengono, da sinistra a destra, il valore della variabile P (pari al valore numerico letto tramite l'istruzione PEEK(197) od alla codifica ASCII del carattere acquisito tramite l'istruzione GET X\$), la lettera corrispondente al tasto premuto, il tipo di istruzione utilizzata per l'acquisizione del segnale, oltre alla solita indicazione circa la direzione selezionata.

Tutte le altre funzioni del programma rimangono invariate, compreso il controllo del colore dello sprite tramite i "pulsanti di fuoco" e la necessità di premere RUN/STOP per interrompere il programma.

Di seguito vengono commentate le istruzioni che differenziano in maniera fondamentale il Programma 2 dal precedente.

Riga
60-70

Commento

Vengono inizializzate le variabili che caratterizzano il menu ridotto.

110

Viene acquisito, mediante l'istruzione PEEK (197), un eventuale segnale proveniente dalla tastiera. Nel caso in cui il contenuto della locazione suddetta sia uguale a 1 o a 60 (vale a dire i valori numerici corrispondenti, rispettivamente, alla pressione del tasto RETURN o della barra spaziatrice) si considera premuto

120

il pulsante di fuoco "virtuale" e si passa alla sezione principale del programma.

Se il valore numerico letto è uguale a 4, corrispondente alla pressione del tasto funzione 'fi', si passa dalla opzione attualmente attiva a quella che la precede, considerando il menu ciclico, con le stesse caratteristiche già illustrate per il programma precedente.

130

Analogamente alla riga precedente. In questo caso il test viene compiuto sul tasto funzione 'f7' (PEEK(197)=3, se premuto) per stabilire se si desidera passare da una voce del menu alla successiva.

230

Se viene attivata la seconda opzione del menu, il programma si dirama verso la sezione opportuna del programma.

250

Acquisizione del segnale tramite istruzione PEEK (197). Viene quindi azzerata una variabile 'flag' FL, che fa da segnale circa la significatività del tasto premuto. Dopo aver richiamato un opportuno sottoprogramma di decodifica, infatti, viene eseguito un test sul valore della variabile FL e, nel caso in cui essa valga 1 (nel caso cioè in cui sia stato premuto un tasto che non fa parte del joystick "virtuale" da noi definito o nessun tasto sia stato premuto), si richiama il sottoprogramma di controllo del colore dello sprite, quindi si ricicla per acquisire un nuovo segnale.

320-350

Gestione del colore dello sprite. In particolare:

330

Viene eseguito un test per stabilire se sia stato premuto uno dei due pulsanti di fuoco definiti sulla tastiera; in caso negativo si esce dal sottoprogramma.

360-370

Controllo del colore dello sprite nel caso in cui il tasto premuto venga acquisito tramite istruzione GET X\$. In particolare:

360

Viene acquisito il tasto premuto. Come potete notare, la lettura avviene senza preventivo svuotamento del buffer, per cui è possibile premere fino a dieci tasti

senza che la memoria tamponi "dimentichi" un segnale della serie. Nel caso in cui non sia stato acquisito alcun segnale, si ricicla immediatamente per attendere la pressione di un tasto, dal momento che non ci sono compiti particolari da svolgere.

N.B. - Questa sezione di acquisizione è comune alla parte del programma che si occupa del joystick "virtuale".

Il carattere alfanumerico acquisito con la istruzione precedente viene convertito nel suo equivalente ASCII; quindi viene compiuto un test per stabilire se sia stato premuto il tasto RETURN (P=13) o la barra spaziatrice (P=32), nel qual caso si richiama il sottoprogramma di gestione del colore dello sprite.

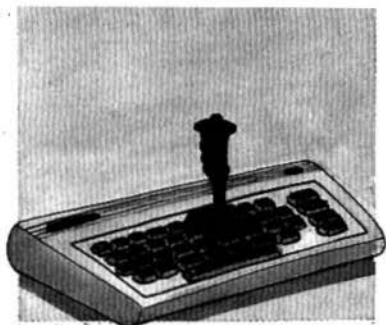
Si azzerla la variabile di 'flag' e si richiama un opportuno sottoprogramma, analogo a quello visto per la decodifica del segnale del joystick "virtuale" acquisito tramite PEEK (197).

770-850

Sottoprogramma di decodifica del segnale acquisito tramite istruzione PEEK (197). I test vengono eseguiti singolarmente, non essendo possibile convertire i valori numerici acquisiti (non consecutivi e spazianti su di un campo di valori piuttosto vario) in modo da poterli utilizzare come indici di eventuali matrici in maniera analoga a quanto fatto per il Programma 1 con la variabile JS. Per ognuno dei valori rilevati il sottoprogramma fornisce il valore SP dell'indice degli sprite, il valore JS dell'indice della matrice contenente le scritte per la "bussola", la variabile alfanumerica X\$, contenente il carattere corrispondente al tasto premuto.

860-940

Analoghe alle righe precedenti. In questo caso il sottoprogramma opera la decodifica partendo dai valori ASCII corrispondenti ai tasti premuti ed acquisiti tramite l'istruzione GET X\$. Anche in questo caso, benché l'ampiezza dell'intervallo



dei valori assunti sia più ristretta, la loro conversione in indici di matrici rende necessario questo meno elegante e più costoso, in termini di tempo, sottoprogramma di decodifica.

Questa riga può essere omessa, dal momento che non viene mai richiamata nel corso del programma. Può però risultare comoda, se si desidera ridefinire alcuni dei tasti utilizzati per le varie funzioni descritte, per conoscerne il valore numerico corrispondente tramite la funzione PEEK (197). I valori numerici dei codici ASCII corrispondenti ai vari tasti possono invece essere reperiti sul "Manuale d'uso del C64", Appendice F.

370

380

```

10 RS="{HOME}{ 25 GIU'":DIMW$(25),SP(10,
2),SP$(10):FORI=1TO25          :rem 193
20 W$(I)=LEFT$(RS,I):NEXT:FORI=1TO10:FORJ
=0TO2:READSP(I,J):NEXT:READSP$(I):NEXT
          :rem 243
30 POKE53280,15:POKE53281,15:PRINT"{CLR}"
:V=53248          :rem 149
40 PRINTCHR$(14):IFPEEK(16004)=4THEN60
          :rem 232
50 FORI=0TO7:FORJ=0TO63:READA:POKE15744+6
4*I+J,A:NEXTJ,I          :rem 231
60 PRINT"{CLR}"TAB(16)"{RED}M E N U'":NS=
2:M$(1)="P{ 2 E}K(197) "          :rem 36
70 M(1,1)=64:M(2,1)=32:M$(2)="GET X$
{ 4 SPAZI}          :rem 78
80 PRINTW$(3)TAB(4)"{BLU}LETTURA TRAMITE
ISTRUZIONE:"          :rem 192
90 FORI=1TO2:PRINTW$(5+I*2)TAB(3)"{RED}
{RVS}"RIGHT$(STR$(I),1)"{OFF}{BLK} - "
M$(I):NEXT          :rem 95
100 PRINTW$(5+NS*2)TAB(7)"{RED}"M$(NS):PR
INTW$(5+VS*2)TAB(7)"{BLK}"M$(VS)
          :rem 255
110 P1=PEEK(197):IFP1=1ORP1=60THEN150
          :rem 169
120 IFP1=4THENVNS=NS:NS=NS-1:IFNS=0THENNS=
2          :rem 247
130 IFP1=3THENVNS=NS:NS=NS+1:IFNS=3THENNS=
1          :rem 247
140 GOTO100          :rem 95
150 PRINT"{CLR}"TAB(2)"{BLU}P"TAB(8)"
{GRN}X$TAB(14)"{BLK}LETTURA IN"TAB(2
8)"{RED}DIREZIONE{BLK}"          :rem 137
160 PRINT"[<A>]{ 3 *}{<S>]{ 2 SPAZI}{<A>}
{ 2 *}{<S>]{ 2 SPAZI}{<A>}{ 10 *}
{<S>]{ 2 SPAZI}{<A>}{ 10 *}{<S>}"
          :rem 160

```

950

```

170 P$=RIGHT$("{ 3 SPAZI}"+STR$(M(NS,1)),
3)          :rem 20
180 PRINT"-{BLU}"P$"{BLK}-{ 2 SPAZI}-
{GRN} 0{BLK}-{ 2 SPAZI}-"M$(NS)"-
{ 2 SPAZI}-{ 10 SPAZI}-"          :rem 62
190 PRINT"[<Z>]{ 3 *}{<X>]{ 2 SPAZI}{<Z>}
{ 2 *}{<X>]{ 2 SPAZI}{<Z>}{ 10 *}
{<X>]{ 2 SPAZI}{<Z>}{ 10 *}{<X>}"FOR
I=1TO10          :rem 175
200 IFSP(I,0)=0THEN220          :rem 234
210 PRINTW$(SP(I,1))TAB(SP(I,2))SP$(I)
          :rem 170
220 NEXT          :rem 211
230 IFNS=2THEN360          :rem 252
250 P=PEEK(197):FL=0:GOSUB770:IFFL=1THENG
OSUB320:GOTO250          :rem 227
260 POKEV+1,121:POKEV+0,156          :rem 27
270 POKEV+23,1:POKEV+29,1:POKEV+39,CO
          :rem 209
280 POKE2040,245+SP:POKEV+21,1:P$=RIGHT$(
STR$(P),3):X$=RIGHT$("{ 2 SPAZI}"+X$,
2)          :rem 56
290 PRINTW$(3)TAB(1)"{BLU}"P$TAB(8)"{GRN}
"X$TAB(28)"{RED}"SP$(JS)          :rem 60
300 IFOS<>0ANDOS<>JSTHENPRINTW$(SP(OS,1))
TAB(SP(OS,2))"{BLK}"SP$(OS)          :rem 31
310 PRINTW$(SP(JS,1))TAB(SP(JS,2))"{RED}"
SP$(JS):OS=JS:GOTO230          :rem 197
320 REM ** COLORE DELLO $SPRITE **
          :rem 44
330 IFP<>1ANDP<>60THENRETURN          :rem 50
340 CO=CO+1:IFCO=15THENCO=0          :rem 128
350 POKEV+39,CO:FORI=1TO500:NEXT:RETURN
          :rem 21
360 GETX$:IFX$=""THEN360          :rem 131
370 P=ASC(X$):IFP=13ORP=32THENGOSUB340:GO
TO360          :rem 106

```

REM-HW

Il Joystick

```
380 FL=0:GOSUB860:IFFL=1THENGOTO360
      :rem 188
390 GOTO260 :rem 109
400 REM ** DATI BUSSOLA ** :rem 91
410 DATA1,7,15,"{ 3 SPAZI}NORD{ 3 SPAZI}
      ",2,17,15,"{ 3 SPAZI}SUD{ 4 SPAZI}",0
      ,0,0,"ILLEGALE" :rem 155
420 DATA3,12,5,"{ 2 SPAZI}OVEST
      { 3 SPAZI}",6,7,5,"NORD_OVEST",5,17,5
      ," SUD_OVEST" :rem 75
430 DATA0,0,0,"ILLEGALE" :rem 200
440 DATA4,12,25,"{ 3 SPAZI}EST{ 4 SPAZI}
      ",7,7,25," NORD_EST ",8,17,25," SUD-
      EST{ 2 SPAZI}" :rem 249
450 REM ** DATA SPRITE NORD ** :rem 73
460 DATA0,8,0,0,28,0,0,62,0,0,127,0,0,255
      ,128,1,255,192,3,255,224,7,255,240
      :rem 108
470 DATA0,127,0,0,127,0,0,127,0,0,127,0,0
      ,127,0,0,255,128,1,255,192,3,247,224
      :rem 195
480 DATA3,227,224,3,193,224,3,128,224,3,0
      ,96,2,0,32,0 :rem 71
490 REM ** DATA SPRITE SUD ** :rem 6
500 DATA2,0,32,3,0,0,96,3,128,224,3,193,224
      ,3,227,224,3,247,224,1,255,192
      :rem 189
510 DATA0,255,128,0,127,0,0,127,0,0,127,0
      ,0,127,0,0,127,0,7,255,240,3,255,224
      :rem 189
520 DATA1,255,192,0,255,128,0,127,0,0,62
      ,0,0,28,0,0,8,0,0 :rem 237
530 REM ** DATA SPRITE OVEST ** :rem 166
540 DATA0,0,0,0,0,0,0,0,0,1,0,0,3,1,248,7
      ,3,240,15,7,224,31,255,192,63,255,128
      :rem 223
550 DATA127,255,0,255,254,0,127,255,0,63
      ,255,128,31,255,192,15,7,224,7,3,240
      :rem 200
560 DATA3,1,248,1,0,0,0,0,0,0,0,0,0,0,0
      :rem 188
570 REM ** DATA SPRITE EST ** :rem 5
580 DATA0,0,0,0,0,0,0,0,0,0,0,128,31,128
      ,192,15,192,224,7,224,240,3,255,248
      :rem 93
590 DATA1,255,252,0,255,254,0,127,255,0,2
      55,254,1,255,252,3,255,248,7,224,240
      :rem 250
600 DATA15,192,224,31,128,192,0,0,128,0,0
      ,0,0,0,0,0,0,0 :rem 216
610 REM ** DATA SPRITE SUD-OVEST **
      :rem 190
620 DATA0,0,0,0,4,0,0,12,0,0,28,0,0,60,0
      ,0,124,0,0,252,0,0,255,240,0,255,224
      :rem 87
630 DATA0,255,192,1,255,128,131,255,0,199
      ,254,0,239,224,0,255,192,0 :rem 12
640 DATA255,128,0,255,0,0,254,0,0,255,0,0
      ,255,128,0,255,192,0,0 :rem 37
650 REM ** DATA SPRITE NORD-OVEST **
      :rem 9
660 DATA0,0,0,255,192,0,255,128,0,255,0,0
      ,254,0,0,255,0,0,255,128,0,255,192,0
      :rem 203
670 DATA239,224,0,199,254,0,131,255,0,1,2
      55,128,0,255,192,0,255,224,0,255,240
      :rem 242
680 DATA0,252,0,0,124,0,0,60,0,0,28,0,0,1
      2,0,0,4,0,0 :rem 215
690 REM ** DATA SPRITE NORD-EST **
      :rem 104
700 DATA0,0,0,0,15,254,0,7,254,0,3,254,0
      ,1,254,0,3,254,0,7,254,0,15,254
      :rem 147
710 DATA0,31,254,1,255,238,3,255,198,7,25
      5,130,15,255,0,31,254,0,0,62,0
      :rem 188
720 DATA0,62,0,0,62,0,0,60,0,0,56,0,0,48
      ,0,0,32,0,0 :rem 173
730 REM ** DATA SPRITE SUD-EST **
      :rem 28
740 DATA0,32,0,0,48,0,0,56,0,0,60,0,0,62
      ,0,0*N
      EWDATA7,255,130,3,255,198,1,255,238,
      0,31,254,0,15,254,0,7,254,0,3,254
      :rem 218
750 DATA0,1,254,0,3,254,0,7,254,0,15,254
      ,0,0,0,0 :rem 56
770 IFP=9THENSP=1:JS=1:X$="W":RETURN
      :rem 160
780 IFP=23THENSP=2:JS=2:X$="X":RETURN
      :rem 208
790 IFP=10THENSP=3:JS=4:X$="A":RETURN
      :rem 185
800 IFP=18THENSP=4:JS=8:X$="D":RETURN
      :rem 193
810 IFP=12THENSP=5:JS=6:X$="Z":RETURN
      :rem 209
820 IFP=62THENSP=6:JS=5:X$="Q":RETURN
      :rem 206
830 IFP=14THENSP=7:JS=9:X$="E":RETURN
      :rem 197
840 IFP=20THENSP=8:JS=10:X$="C":RETURN
      :rem 234
850 FL=1:RETURN :rem 183
860 IFP=87THENSP=1:JS=1:X$="W":RETURN
      :rem 214
870 IFP=88THENSP=2:JS=2:X$="X":RETURN
      :rem 219
880 IFP=65THENSP=3:JS=4:X$="A":RETURN
      :rem 195
890 IFP=68THENSP=4:JS=8:X$="D":RETURN
      :rem 207
900 IFP=90THENSP=5:JS=6:X$="Z":RETURN
      :rem 215
910 IFP=81THENSP=6:JS=5:X$="Q":RETURN
      :rem 207
920 IFP=69THENSP=7:JS=9:X$="E":RETURN
      :rem 207
930 IFP=67THENSP=8:JS=10:X$="C":RETURN
      :rem 245
940 FL=1:RETURN :rem 183
950 PRINT"{CLR}":FORI=1TO1000:PRINT"
      {HOME}"RIGHT$(("{ 4 SPAZI}"+STR$(PEEK(
      197)),3):NEXT :rem 27
```


OK-List per la perfetta battitura dei listati

Inserite in memoria l'OK-LIST prima di accingervi a battere i listati pubblicati: avrete a disposizione un formidabile mezzo per essere sicuri di aver inserito nel computer ogni carattere in modo assolutamente corretto.

Tutti i nostri listati sono caratterizzati da un numero aggiunto ad ogni linea di programma, ad esempio: rem 123. NON COPIATE QUESTA PARTE DELLA LINEA! Serve unicamente per vostra informazione. L'istruzione REM ha l'unico scopo di renderla inoffensiva se per errore doveste batterla.

Se, prima di iniziare il lavoro di battitura del listato, caricate in memoria OK-LIST e lo attivate con RUN, per ogni linea che battete verrà visualizzato un numero (detto checksum) nell'angolo superiore sinistro dello schermo.

Confrontate questo numero con quello pubblicato per ogni linea del listato: se non corrisponde, ciò significa che avete commesso un errore nella battitura della linea stessa.

Le ore buttate per cercare e correggere errori di battitura in programmi che non funzionano saranno solo un lontano ricordo!

Due ultime cose di cui tenere conto: OK-LIST non tiene conto degli spazi: questo per vostra convenienza, poiché questi generalmente sono poco importanti.

Infine, OK-LIST è allocato nel buffer del registratore (locazioni 886-1018), quindi prima di salvare su cassetta il programma che state battendo, ricordatevi di disabilitarlo premendo contemporaneamente RUN/STOP + RESTORE. Potrete poi riattivarlo con SYS 886.

Il sistema più comodo per usare OK-LIST è quello di battere normalmente il programma, quindi listarlo e portarsi con il cursore sul numero di linea più basso che appare sul video; premere quindi RETURN e confrontare il numero di checksum visualizzato nell'angolo superiore sinistro del video con quello pubblicato per la data riga del programma. Il cursore si porterà automaticamente sulla linea successiva, e non dovrete fare altro che premere il checksum seguente.

Data l'area di memoria in cui si trova, OK-LIST non può essere usato per controllare un programma già salvato su cassetta, mentre ciò è possibile se il programma è stato registrato su disco.



OK-LIST versione per VIC 20 e C 64

```

100 PRINT"{CLR}ATTENDERE PREGO..."
105 FORI=886TO1018:READA:CK=CK+A:POKEI,A:
NEXT
110 IF CK<>17539 THEN PRINT"{GIU'}ERRORE
NELLE ISTRUZIONI DATA":END
120 SYS886:PRINT"{CLR}{ 2 GIU'}OK-LIST AT
TIVATO.":NEW
886 DATA 173,036,003,201,150,208
892 DATA 001,096,141,151,003,173
898 DATA 037,003,141,152,003,169
904 DATA 150,141,036,003,169,003
910 DATA 141,037,003,169,000,133
916 DATA 254,096,032,087,241,133
922 DATA 251,134,252,132,253,008
928 DATA 201,013,240,017,201,032
934 DATA 240,005,024,101,254,133
940 DATA 254,165,251,166,252,164
946 DATA 253,040,096,169,013,032
952 DATA 210,255,165,214,141,251
958 DATA 003,206,251,003,169,000
964 DATA 133,216,169,019,032,210
970 DATA 255,169,018,032,210,255
976 DATA 169,058,032,210,255,166
982 DATA 254,169,000,133,254,172
988 DATA 151,003,192,087,208,006
994 DATA 032,205,189,076,235,003
1000 DATA 032,205,221,169,032,032
1006 DATA 210,255,032,210,255,173
1012 DATA 251,003,133,214,076,173
1018 DATA 003
  
```

MLX per VIC 20

e C64 (versione 2.0 per C64)

di C. Brannon
trad. e adatt.
di M. Cristub Grizzi
e F. Stella

M LX è un programma che permette di inserire listati in linguaggio macchina esenti in modo assoluto da errori e senza la perdita di tempo del dover battere e controllare lunghe sequenze di istruzioni DATA.

Molti dei nostri listati di programmi in linguaggio macchina hanno il formato MLX (li riconoscete dal fatto che sono esclusivamente numerici) e richiedono quindi che MLX sia caricato in memoria ed eseguito prima della battitura del listato.

MLX vi chiederà l'indirizzo della locazione di partenza e quello della locazione finale del programma da caricare. Questi valori sono sempre indicati nell'articolo che accompagna il listato. MLX vi segnala automaticamente gli errori di battitura MENTRE STATE DIGITANDO IL LISTATO e vi chiede di reinserire la linea errata. L'ultimo numero battuto di ogni

linea rappresenta il checksum e viene visualizzato in reverse.

Sono inoltre disponibili altri comandi, quali: SHIFT-N nuovo indirizzo: permette di cambiare l'indirizzo della linea che volete battere ed è utile nel caso si inseriscano i listati in più parti.

SHIFT-D display: lista i dati inseriti tra due indirizzi di inizio e fine.

SHIFT-L load: carica un file da nastro o SHIFT-S save: salva su nastro o disco un file in formato MLX.

Una volta battuto il listato, e salvato tramite MLX, si potrà caricare direttamente il programma con un'istruzione LOAD "nome del file", 1,1 per il registratore, oppure LOAD "nome del file", 8,1 per l'unità a dischi, salvo diverse indicazioni date negli articoli corrispondenti ai vari programmi. Il programma sarà generalmente mandato in esecuzione con una SYS (indirizzo di partenza).

La versione originaria di MLX per C64 ha subito, dalla prima pubblicazione, diversi ritocchi e miglioramenti, fino alla versione 2.0 qui presentata.

È stato in primo luogo modificato il colore di fondo dello schermo e del bordo, in modo da risultare meno stancante alla vista; sono state ulteriormente perfezionate le routine in linguaggio macchina di salvataggio e caricamento dei programmi; è cambiata la forma del cursore e, cosa più importante, è stato aggiunto un tastierino numerico per gli utenti che si trovano più a loro agio con una diversa disposizione dei tasti.

Oltre a poter usare i soliti tasti numerici, sono stati ridefiniti alcuni tasti alfabetici, in modo da generare ugualmente dei numeri. I tasti ridefiniti sono i seguenti:

U I O	7 8 9
H J K L	diventano 0 4 5 6
M , .	1 2 3

Le persone abituate ad usare tali tastierini accresceranno senz'altro, in brevissimo tempo, la velocità di battitura dei programmi in linguaggio macchina.

MLX versione per VIC 20

```

100 PRINT"{CLR}{PUR}";CHR$(142);CHR$(8);
:rem 181
101 POKE788,194:REM DISABILITA RUN/STOP
:rem 144
110 PRINT"{RVS}{ 14 SPAZI}" :rem 117
120 PRINT"{RVS}{DES}{OFF}[<*>]E{RVS}
{DES}{DES}{ 2 SPAZI}[<*>]{OFF}[<*>]
E{RVS}E{RVS}" :rem 191
130 PRINT"{RVS}{DES}{<G>}{DES}
{ 2 DES}{OFF}E{RVS}E[<*>]{OFF}[<*>]
{RVS}{OFF}" :rem 122
140 PRINT"{RVS}{ 14 SPAZI}" :rem 120
200 PRINT"{ 2 GIU'J}{PAU}{BLK}UN PROGRAMMA
PER L'IN-TRODUZIONE{ 2 SPAZI}DI ROUT
INEIN LINGUAGGIO MACCHINA"; :rem 202
205 PRINT"A PROVA DI ERRORE{ 5 GIU'}"
:rem 226
210 PRINT"{BLK}{ 4 SU}INDIRIZZO DI PARTEN
ZA":INPUTS:F=1-F:C$=CHR$(31+119*F)
:rem 52
220 IFS<256ORS>32767THENGOSUB3000:GOTO210
:rem 2
225 PRINT:PRINT:PRINT:PRINT :rem 123
230 PRINT"{BLK}{ 3 SU}INDIRIZZO CONCLUSIV
O":INPUTE:F=1-F:C$=CHR$(31+119*F)
:rem 170
240 IFE<256ORE>32767THENGOSUB3000:GOTO230
:rem 234
250 IFE<STHENPRINTC$;"{RVS}INDIRIZZO CONC
LUSIVO <INDIRIZZO DI PARTENZA
{ 3 GIU' }":GOSUB1000:GOTO230:rem 27
260 PRINT:PRINT:PRINT :rem 179
300 PRINT"{CLR}";CHR$(14):AD=S :rem 56
310 PRINTRIGHT$( "0000"+MID$(STR$(AD),2),5
);";":FORJ=1TO6 :rem 234
320 GOSUB570:IFN=-1THENJ=J+N:GOTO320
:rem 228
390 IFN=-211THEN710 :rem 62
400 IFN=-204THEN790 :rem 64
410 IFN=-206THENPRINT:PRINT"GIU'INSERIS
CI UN NUOVO{ 4 SPAZI}INDIRIZZO";:INPU
TZZ :rem 42

```

```

415 IFN=-206THENIFZZ<SORZZ>ETHENPRINT"
{RVS}ESCE DAL CAMPO DI VA-" :rem 150
416 IFN=-206THENIFZZ<SORZZ>ETHENPRINT"
{RVS}LORI INDICATO":GOSUB1000:GOTO410
:rem 105
417 IFN=-206THENAD=ZZ:PRINT:GOTO310
:rem 238
420 IFN<>-196THEN480 :rem 133
430 PRINT:INPUT"LISTATO DA";F:PRINT,"
{ 2 SIN}A";:INPUTT :rem 29
440 IFF<SORF>EORT<SORT>ETHENPRINT"MINIMO"
;S;"{SIN}, MASSIMO";E:GOTO430 :rem 36
450 FORI=FTOTSTEP6:PRINT:PRINTRIGHT$( "000
0"+MID$(STR$(I),2),5);";": :rem 30
455 FORK=0TO5:N=PEEK(I+K):IFK=3THENPRINTS
PC(10); :rem 34
457 PRINTRIGHT$( "00"+MID$(STR$(N),2),3);"
,"; :rem 157
460 GETA$:IFA$>"THENPRINT:PRINT:GOTO310
:rem 25
470 NEXTK:PRINTCHR$(20);:NEXTI:PRINT:PRIN
T:GOTO310 :rem 50
480 IFN<0THENPRINT:GOTO310 :rem 168
490 A(J)=N:NEXTJ :rem 199
500 CKSUM=AD-INT(AD/256)*256:FORI=1TO6:CK
SUM=(CKSUM+A(I))AND255:NEXT :rem 200
510 PRINTCHR$(18);:GOSUB570:PRINTCHR$(20)
:rem 234
515 IFN=CKSUMTHEN530 :rem 255
520 PRINT:PRINT"LA RIGA E' STATA INSE-RI
T A IN MANIERA ERRATA" :rem 39
525 PRINT"RI-INSEIRILA":PRINT:GOSUB1000:G
OTO310 :rem 167
530 GOSUB2000 :rem 218
540 FORI=1TO6:POKEAD+I-1,A(I):NEXT
:rem 80
550 AD=AD+6:IFAD<ETHEN310 :rem 212
560 GOTO710 :rem 108
570 N=0:Z=0 :rem 88

```

```

580 PRINT "[<+>]"; :rem 79
581 GETA$:IFAS=" " THEN 581 :rem 95
585 PRINT CHR$(20);:A=ASC(AS):IFA=13ORA=44 :rem 229
ORA=32THEN 670 :rem 229
590 IFA>128THEN --A:RETURN :rem 137
600 IFA>20THEN 630 :rem 10
610 GOSUB 690:IFI=1ANDT=44THEN --1:PRINT" :rem 172
{SIN}{SIN}";:GOTO 690 :rem 109
620 GOTO 570 :rem 109
630 IFA<48ORA>57THEN 580 :rem 105
640 PRINT A$;:N=N*10+A-48 :rem 106
650 IFN>255THEN A=20:GOSUB 1000:GOTO 600 :rem 229
660 Z=Z+1:IFZ<3THEN 580 :rem 71
670 IFZ=0THEN GOSUB 1000:GOTO 570 :rem 114
680 PRINT " ";:RETURN :rem 240
690 S%=PEEK(209)+256*PEEK(210)+PEEK(211) :rem 149
:rem 68
692 FOR I=1 TO 3:T=PEEK(S%-I) :rem 68
695 IFT<>44ANDT<>58THEN POKES%-I,32:NEXT :rem 205
700 PRINT LEFT$(" { 3 SIN}",I-1);:RETURN :rem 7
710 PRINT "{CLR}{RVS}*** SAVE *** :rem 236
{ 3 GIU' }"
720 INPUT "{GIU' } NOME DEL FILE";F$: :rem 199
:rem 128
730 PRINT:PRINT "{ 2 GIU' } {RVS}N{OFF}ASTRO :rem 128
O {RVS}D{OFF}ISCO: (N/D)"
740 GETA$:IFAS<>"N"ANDA$<>"D"THEN 740 :rem 30
750 DV=1-7*(AS="D"):IFDV=8THEN F$="0:"+F$ :rem 158
760 T$=F$:ZK=PEEK(53)+256*PEEK(54)-LEN(T$ :rem 3
):POKE 782,ZK/256
762 POKE 781,ZK-PEEK(782)*256:POKE 780,LEN( :rem 109
(T$):SYS65469
763 POKE 780,1:POKE 781,DV:POKE 782,1:SYS654 :rem 69
66
765 POKE 254,S/256:POKE 253,S-PEEK(254)*256 :rem 12
:POKE 780,253
766 POKE 782,E/256:POKE 781,E-PEEK(782)*256 :rem 124
:SYS65496
770 IF (PEEK(783)AND 1)OR (ST AND 191)THEN 780 :rem 111

```

```

775 PRINT "{GIU' } OPERAZIONE CONCLUSA.":END :rem 152
780 PRINT "{GIU' } ERRORE DI REGISTRAZIONE- :rem 8
RIPROVA.":IFDV=1THEN 720
781 OPEN 15,8,15:INPUT #15,E1$,E2$:PRINT E1$ :rem 103
;E2$:CLOSE 15:GOTO 720
782 GOTO 720 :rem 115
790 PRINT "{CLR}{RVS}*** LOAD *** :rem 212
{ 2 GIU' }"
800 INPUT "{ 2 GIU' } NOME DEL FILE";F$: :rem 215
810 PRINT:PRINT "{ 2 GIU' } {RVS}N{OFF}ASTRO :rem 127
O {RVS}D{OFF}ISCO: (N/D)"
820 GETA$:IFAS<>"N"ANDA$<>"D"THEN 820 :rem 28
830 DV=1-7*(AS="D"):IFDV=8THEN F$="0:"+F$ :rem 157
840 T$=F$:ZK=PEEK(53)+256*PEEK(54)-LEN(T$ :rem 2
):POKE 782,ZK/256
841 POKE 781,ZK-PEEK(782)*256:POKE 780,LEN( :rem 107
(T$):SYS65469
845 POKE 780,1:POKE 781,DV:POKE 782,1:SYS654 :rem 70
66
850 POKE 780,0:SYS65493 :rem 11
860 IF (PEEK(783)AND 1)OR (ST AND 191)THEN 870 :rem 111
865 PRINT "{GIU' } OPERAZIONE CONCLUSA.":GOT :rem 142
O 310
870 PRINT "{GIU' } ERRORE DI CARICAMENTO-RI :rem 43
ROVA. {GIU' }":IFDV=1THEN 800
880 OPEN 15,8,15:INPUT #15,E1$,E2$:PRINT E1$ :rem 102
;E2$:CLOSE 15:GOTO 800
1000 REM CICALINO :rem 231
1001 POKE 36878,15:POKE 36874,190 :rem 206
1002 FORW=1 TO 300:NEXT W :rem 117
1003 POKE 36878,0:POKE 36874,0:RETURN :rem 74
2000 REM CAMPANELLO :rem 130
2001 FORW=15 TO 0 STEP -1:POKE 36878,W:POKE 368 :rem 22
76,240:NEXT W
2002 POKE 36876,0:RETURN :rem 119
3000 PRINT C$;" {RVS}NON IN PAGINA ZERO O :rem 119
{ 2 SPAZI } SU ROM { 4 GIU' }":GOTO 1000 :rem 23

```

MLX versione 2.0 per C64

```

100 PRINT "{CLR}{CYN}";CHR$(142);CHR$(8);: :rem 71
POKE 53280,0:POKE 53281,0
101 POKE 788,52:REM DISABILITA RUN/STOP :rem 89
110 PRINT "{RVS}{ 40 SPAZI}"; :rem 176
120 PRINT "{RVS}{ 15 SPAZI } {DES} {OFF} {<*>} :rem 176
{RVS} {DES} {DES} { 2 SPAZI } {<*>} {OFF}
{<*>} {RVS} {RVS} { 13 SPAZI}"; :rem 250
130 PRINT "{RVS}{ 15 SPAZI } {DES} [ <N> ] [ <H> ] :rem 220
{DES} { 2 DES } {OFF} {RVS} {<*>} {OFF}
{<*>} {RVS} { 13 SPAZI}";
140 PRINT "{RVS}{ 40 SPAZI}"; :rem 120
200 PRINT "{ 2 GIU' } {PUR} EDITOR DI CODICE :rem 126
MACCHINA VERSIONE 2.0 { 3 GIU' }"
210 PRINT "[<5>] { 2 SU } INDIRIZZO DI PARTE :rem 62
NZA { 2 SPAZI}";:INPUTS:F=1-F:C$=CHR$(
31+119*F) :rem 64
220 IFS<256OR(S>40960ANDS<49152)ORS>53247 :rem 235
THEN GOSUB 3000:GOTO 210
225 PRINT:PRINT:PRINT :rem 180

```

```

230 PRINT "[<5>] { 2 SU } INDIRIZZO CONCLUSI :rem 92
VO { 3 SPAZI}";:INPUTS:F=1-F:C$=CHR$(3
1+119*F)
240 IFE<256OR(E>40960ANDE<49152)ORE>53247 :rem 183
THEN GOSUB 3000:GOTO 230
250 IFE<STHEN PRINT C$;" {RVS} INDIRIZZO CONC :rem 200
LUSIVO<INDIRIZZO INIZIALE"
255 IFE<STHEN GOSUB 1000:GOTO 230 :rem 119
260 PRINT:PRINT:PRINT :rem 179
300 PRINT "{CLR}";CHR$(14):AD=S:POKEV+21,0 :rem 225
310 A=1:PRINT RIGHT$("0000"+MID$(STR$(AD), :rem 227
2),5);:FORJ=ATO6 :rem 33
315 FORJ=ATO6 :rem 33
320 GOSUB 570:IFN=-1THENJ=J+N:GOTO 320 :rem 228
390 IFN=-211THEN 710 :rem 62
400 IFN=-204THEN 790 :rem 64
410 IFN=-206THEN PRINT:INPUT "{GIU' } INSE :rem 13
RE IL NUOVO INDIRIZZO";ZZ
414 IFN=-206THEN IFZZ<SORZZ>ETHEN PRINT" :rem 105
{RVS}ESCE DAL CAMPO DI VALORI INDICAT
O"

```

```

415 IFN=-206THENIFZZ<SORZZ>ETHENGOSUB1000
:GOTO410 :rem 202
417 IFN=-206THENAD=ZZ:PRINT:GOTO310
:rem 238
420 IFN<>-196THEN480 :rem 133
430 PRINT:INPUT"LISTATO:DA";F:PRINT"
{ 9 SPAZI}A";:INPUTT :rem 183
440 IFF<SORF>EORT<SORT>ETHENPRINT"MINIMO"
;S;" MASSIMO";E;"! [<5>":GOTO430
:rem 147
450 FORI=FTOTSTEP6:PRINT:PRINTRIGHTS("000
0"+MID$(STR$(I),2),5);":":rem 30
451 FORK=0TO5:N=PEEK(I+K):PRINTRIGHTS("00
"+MID$(STR$(N),2),3);":":rem 66
460 GETA$:IFA$>"THENPRINT:PRINT:GOTO310
:rem 25
470 NEXTK:PRINTCHR$(20);:NEXTI:PRINT:PRIN
T:GOTO310 :rem 50
480 IFN<0THENPRINT:GOTO310 :rem 168
490 A(J)=N:NEXTJ :rem 199
500 CKSUM=AD-INT(AD/256)*256:FORI=1TO6:CK
SUM=(CKSUM+A(I))AND255:NEXT :rem 200
510 PRINTCHR$(18);:GOSUB570:PRINTCHR$(146
); :rem 94
511 IFN=-1THENA=6:GOTO315 :rem 254
515 PRINTCHR$(20):IFN=CKSUMTHEN530
:rem 122
520 PRINT:PRINT"{RED}LA LINEA E' STATA IN
SERITA IN MANIERA" :rem 157
525 PRINT"ERRATA. RIPETERE [<5>":PRINT:GO
SUB1000:GOTO310 :rem 27
530 GOSUB2000 :rem 218
540 FORI=1TO6:POKEAD+I-1,A(I):NEXT:POKE54
272,0:POKE54273,0 :rem 227
550 AD=AD+6:IFAD<ETHEN310 :rem 212
560 GOTO710 :rem 108
570 N=0:Z=0 :rem 88
580 PRINT" [<->": :rem 81
581 GETA$:IFA$="THEN581 :rem 95
582 AV=- (A$="M")-2*(A$="")-3*(A$=".")-4*
(A$="J")-5*(A$="K")-6*(A$="L") :rem 41
583 AV=AV-7*(A$="U")-8*(A$="I")-9*(A$="O"
):IFA$="H"THENA$="0" :rem 134
584 IFAV>0THENA$=CHR$(48+AV) :rem 134
585 PRINTCHR$(20);:A=ASC(A$):IFA=13ORA=44
ORA=32THEN670 :rem 229
590 IFA>128THENN=-A:RETURN :rem 137
600 IFA<>20THEN630 :rem 10
610 GOSUB690:IFI=1ANDT=44THENN=-1:PRINT"
{SIN} {SIN}";:GOTO690 :rem 172
620 GOTO570 :rem 109
630 IFA<48ORA>57THEN580 :rem 105
640 PRINTA$:N=N*10+A-48 :rem 106
650 IFN>255THENA=20:GOSUB1000:GOTO600
:rem 229
660 Z=Z+1:IFZ<3THEN580 :rem 71
670 IFZ=0THENGOSUB1000:GOTO570 :rem 114
680 PRINT",";:RETURN :rem 240
690 S%=PEEK(209)+256*PEEK(210)+PEEK(211)
:rem 149
691 FORI=1TO3:T=PEEK(S%-I) :rem 67
695 IFT<>44ANDT<>58THENPOKES%-I,32:NEXT
:rem 205
700 PRINTLEFTS(" { 3 SIN}",I-1);:RETURN
@I :rem 7
710 PRINT"{CLR}{RVS}*** SAVE ***
{ 3 GIU'": :rem 236
715 PRINT"{ 2 GIU'}PREMERE {RVS}RETURN
{OFF} PER USCIRE DAL SAVE{GIU'":
:rem 103

```

```

720 F$="":INPUT"GIU'NOME DEL FILE";F$:I
FF$=""THENPRINT:PRINT:GOTO310
:rem 42
730 PRINT:PRINT" { 2 GIU'}{RVS}N{OFF}ASTRO
O {RVS}D{OFF}ISCO: (N/D)":rem 128
740 GETA$:IFA$<"N"ANDAS<"D"THEN740
:rem 30
750 DV=1-7*(A$="D"):IFDV=8THENF$="0":"+F$:
OPEN15,8,15,"S"+F$:CLOSE15 :rem 212
760 T$=F$:ZK=PEEK(53)+256*PEEK(54)-LEN(T$
):POKE782,ZK/256 :rem 3
762 POKE781,ZK-PEEK(782)*256:POKE780,LEN(
T$):SYS65469 :rem 109
763 POKE780,1:POKE781,DV:POKE782,1:SYS654
66 :rem 69
765 K=S:POKE254,K/256:POKE253,K-PEEK(254)
*256:POKE780,253 :rem 17
766 K=E+1:POKE782,K/256:POKE781,K-PEEK(78
2)*256:SYS65496 :rem 235
770 IF(PEEK(783)AND1)OR(191ANDST)THEN780
:rem 111
775 PRINT"GIU' FATTO. {GIU'":GOTO310
:rem 201
780 PRINT"GIU'ERRORE NEL SAVE-RIPROVA!"
:IFDV=1THEN720 :rem 104
781 OPEN15,8,15:INPUT#15,E1$,E2$:PRINT#1$
;E2$:CLOSE15:GOTO720 :rem 103
790 PRINT"{CLR}{RVS}*** LOAD ***
{ 2 GIU'": :rem 212
795 PRINT"{ 2 GIU'}PREMERE {RVS}RETURN
{OFF} PER USCIRE DAL LOAD{GIU'":
:rem 96
800 F$="":INPUT"{ 2 GIU'}NOME DEL FILE";F
$:IFF$=""THENPRINT:GOTO310 :rem 115
810 PRINT:PRINT" { 2 GIU'}{RVS}N{OFF}ASTRO
O {RVS}D{OFF}ISCO: (N/D)":rem 127
820 GETA$:IFA$<"N"ANDAS<"D"THEN820
:rem 28
830 DV=1-7*(A$="D"):IFDV=8THENF$="0":"+F$
:rem 157
840 T$=F$:ZK=PEEK(53)+256*PEEK(54)-LEN(T$
):POKE782,ZK/256 :rem 2
841 POKE781,ZK-PEEK(782)*256:POKE780,LEN(
T$):SYS65469 :rem 107
845 POKE780,1:POKE781,DV:POKE782,1:SYS654
66 :rem 70
850 POKE780,0:SYS65493 :rem 11
860 IF(PEEK(783)AND1)OR(191ANDST)THEN870
:rem 111
865 PRINT"GIU' FATTO.":GOTO310 :rem 184
870 PRINT"GIU'ERRORE NEL LOAD-RIPETI!
{GIU'":IFDV=1THEN800 :rem 19
880 OPEN15,8,15:INPUT#15,E1$,E2$:PRINT#1$
;E2$:CLOSE15:GOTO800 :rem 102
1000 REM CICALINO :rem 231
1001 POKE54296,15:POKE54277,45:POKE54278,
165 :rem 207
1002 POKE54276,33:POKE54273,6:POKE54272,5
:rem 42
1003 FORT=1TO200:NEXT:POKE54276,32:POKE54
273,0:POKE54272,0:RETURN :rem 202
2000 REM CAMPANELLO :rem 130
2001 POKE54296,15:POKE54277,0:POKE54278,2
47 :rem 152
2002 POKE54276,17:POKE54273,40:POKE54272,
0 :rem 86
2003 FORT=1TO100:NEXT:POKE54276,16:RETURN
:rem 57
3000 PRINTC$;"{RVS} NON IN PAGINA ZERO O
SU{DES}ROM ":GOTO1000 :rem 240

```



Superturbo per VIC 20 e C64

di **H. De Ceukelaire**
trad. e adatt.
di **M. Cristuib Grizzi**

SUPERCOMMODORE è lieta di presentare ai suoi lettori uno dei più interessanti programmi fino ad ora pubblicati: benché sembri impossibile, questa poderosa utility permette al registratore a cassette di salvare, caricare e verificare un programma con la stessa velocità del disk drive!

Si noti bene che non è il solito ed ormai diffuso "Turbotape" per C64, ma consiste in un programma ben più potente, in quanto, una volta salvato su nastro un file in formato "turbo" e volendolo ricaricare il giorno dopo, esso si caricherà in memoria con un semplice LOAD, senza più dover avere in memoria SUPERTURBO.

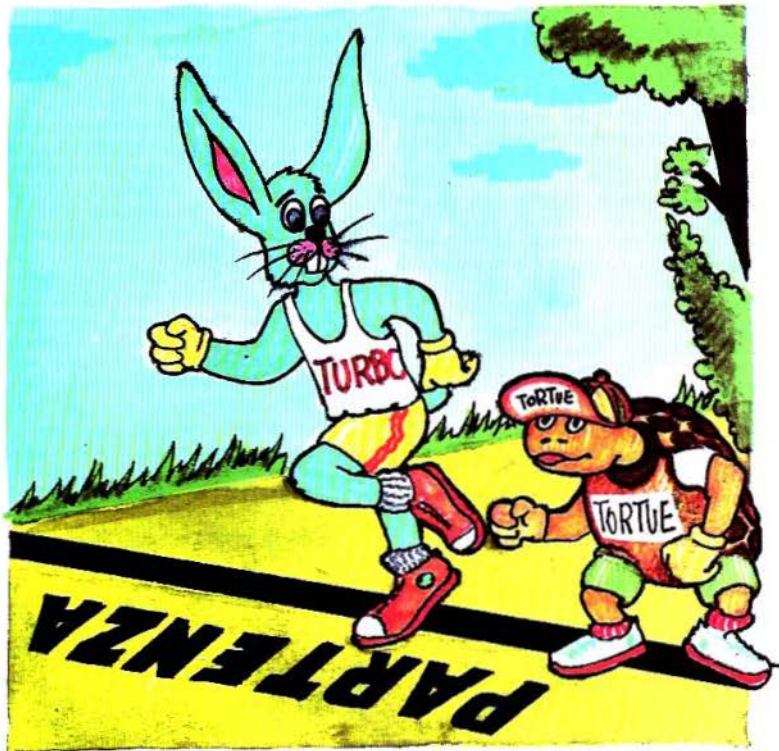
Per il salvataggio si digita semplicemente TSAVE invece di SAVE, ed il computer trasferisce il programma su nastro alla velocità della luce...

Anche dopo aver usato SUPERTURBO per intere settimane, stenterete ancora a credere che sia possibile salvare e caricare così velocemente un programma da cassetta!

SUPERTURBO è in versione per C64 e per VIC 20 (è richiesta l'espansione di memoria da almeno 8 Kbyte per ospitare il generatore del SUPERTURBO, mentre una volta generato il programma questo funzionerà anche sul VIC inespanso).

Esistono veramente poche regole assolute nell'uso del computer, ed una di esse è che, da che mondo è mondo, i registratori a cassette sono sempre stati enormemente più lenti dei drive per floppy disk. Lunghi programmi, che impiegano solamente pochi secondi per essere caricati in memoria da disco, richiedono lunghe attese di interi minuti se caricati da nastro. Ciò avveniva fino a ieri.

SUPERTURBO riscrive queste regole assolute. SUPERTURBO è una utility che porta la velocità del vostro registratore ai livelli di quella del disk drive. Non richiede alcuna modifica né al vostro computer né al registratore di cui disponete e funziona sia con programmi BASIC che con programmi in linguaggio



macchina. Soprattutto, è "user friendly", cioè molto facile da usare anche per i principianti. Il grande vantaggio di SUPERTURBO, rispetto ad altri programmi concorrenti, è che esso non richiede di ricaricare in memoria l'utility prima di eseguire il "Turboload" di un programma.

Tutto ciò che dovete fare è leggere le seguenti istruzioni d'uso.

Digitare e personalizzare il SUPERTURBO

Le seguenti note per la digitazione del SUPERTURBO si rivolgono, ovviamente, a chi riceve la rivista senza cassetta allegata; raccomandiamo tuttavia a tutti di leggere questo paragrafo, in quanto verranno descritte alcune POKE da ef-

fettuare prima di caricare il programma (anche quindi dalla cassetta allegata a SUPERCOMMODORE) e come "personalizzare" il programma.

SUPERTURBO è scritto interamente in linguaggio macchina. Il programma BASIC qui presentato nelle due versioni per VIC 20 e C64, e inserito sulla cassetta allegata alla rivista, serve per creare una copia del SUPERTURBO sia su disco che su nastro.

Raccomandiamo vivamente, a chi non fosse in possesso della nostra cassetta, di digitalarlo con la massima attenzione e di usare OK-LIST (in altra parte della rivista); come sempre, ed a maggior ragione in questo caso, salvate il programma dopo averlo digitato e prima di dargli il RUN (dal momento che il program-

ma modifica, durante l'esecuzione, importanti puntatori della memoria). Con questa procedura, se un errore di battitura provoca il blocco del computer, lo si potrà tranquillamente spegnere e riaccendere e ricaricare il programma per ricercare e correggere l'errore. Dal momento che i dati del SUPERTURBO vengono inseriti nell'area normalmente occupata dai programmi BASIC, occorre rimappare la memoria prima di caricare il programma generatore del SUPERTURBO. Per il C64 occorre spegnere e riaccendere il computer e digitare in modo diretto:

POKE44,14:POKE14+256,0:NEW

Quindi occorre premere RETURN e cedere al caricamento del generatore. Per il VIC20 occorre spegnere il computer, inserire l'espansione di memoria da almeno 8 Kbyte, accenderlo e digitare in modo diretto:

POKE44,32:POKE32+256,0:NEW
POKE648,30:SYS58648

Prima di mandare in esecuzione il programma - sia in versione VIC 20 che in versione C64 - controllate la linea 10: la variabile di stringa FI\$ determina il nome con il quale verrà salvata la copia del SUPERTURBO. Modificatela, se preferite un nome differente. Inoltre, se desiderate creare una copia del SUPERTURBO su disco invece che su nastro, modificate - sempre in linea 10 - la D=1 in D=8. Assicuratevi che la cassetta o il disco sul quale desiderate registrare il SUPERTURBO sia correttamente inserito, rispettivamente nel registratore o nel drive, prima di dare il RUN al programma generatore. Una volta usato quest'ultimo per creare una copia del SUPERTURBO su nastro o disco, esso non vi servirà più. La versione del SUPERTURBO che avete creato, chiamata "TURBO/64" o "TURBO/VIC" (a meno che abbiate modificato la linea 10), può essere caricata ed eseguita come un normale programma BASIC. Non è necessario usare l'indirizzo secondario 1 (es. LOAD "TURBO/64", 8, 1 o LOAD "TURBO/64", 1, 1) quando si carica il SUPERTURBO. Una volta creato, il SUPERTURBO può essere caricato ed eseguito sul Vic 20 in qualsiasi configurazione di memoria.

Facile da usarsi

Ecco alcune delle caratteristiche principali del SUPERTURBO:

- si carica automaticamente al di fuori dell'area di memoria normalmente utilizzata dai programmi BASIC;
- protegge qualsiasi configurazione di memoria stiate usando. Solo durante il TURBOSAVE ed il TURBOVERIFY la ROM del C64 viene passata in RAM; alla fine di queste operazioni viene ripristinata la configurazione precedente;
- SUPERTURBO può essere usato con altre utility, come il SIMON'S BA-

SIC, il SUPERMONITOR ed il PAL;

- SUPERTURBO trasferisce con sicurezza anche programmi molto lunghi (fino a 49 Kbyte di lunghezza nel C64). SUPERTURBO sottrae al C64 solo 642 byte di RAM, e 639 nel VIC 20;
- i nomi dei file possono avere la lunghezza usuale di 16 caratteri;
- oltre a trasferire programmi BASIC, SUPERTURBO salva, carica o verifica dati o codice macchina da qualunque parte di RAM si desidera, con l'eccezione della RAM nascosta "sotto" la ROM del Kernal. Può essere invece salvata anche la RAM "sotto" la ROM del BASIC nel C64;

- un normale comando LOAD caricherà in memoria un qualsiasi programma salvato con SUPERTURBO ed il caricamento avverrà a velocità "Turbo". È molto semplice usare il SUPERTURBO: spegnete e riaccendete il computer; se desiderate usare utility addizionali, come ad esempio il SIMON'S BASIC, caricate e mandate in esecuzione per prima cosa queste ultime. Quindi digitate NEW. Ora caricate il SUPERTURBO e mandatelo in esecuzione; nella versione per C64 apparirà un menu che offre due tipi di locazione a scelta per il SUPERTURBO:

1. Nella RAM del BASIC. L'indirizzo finale del SUPERTURBO qui rilocato sarà correntemente indicato come limite della memoria dal puntatore nelle locazioni 55 e 56. Si deve scegliere questa opzione se volete far coesistere il SUPERTURBO con alcune utility. Il SIMON'S BASIC, ad esempio, è una di queste. Qualunque utility che faccia uso della RAM tra 52606 e 53247 (per esempio, il DOS WEDGE) richiede questa opzione. (Questa è invece l'unica opzione possibile per il VIC 20 ed avviene in modo automatico).

2. Nel buffer di 4 Kbyte. Usando questa opzione, è possibile "turbosavare" tutta la RAM tra 0 e 52606 in un unico lungo blocco, che comprende la RAM "sotto" la ROM.

Per prendere confidenza con l'uso del SUPERTURBO, per il momento evitiamo di usarlo in combinazione con altre utility.

Accendete semplicemente il computer, caricate SUPERTURBO e dategli il RUN (non cercate di modificare in alcun modo la parte in BASIC del SUPERTURBO!). Per il C64 selezionate l'opzione 2.

Utilizzo del SUPERTURBO

Leggerete ora sullo schermo dove è stato rilocato il SUPERTURBO, insieme ai comandi da usare per attivare le caratteristiche del programma. Digitate NEW per eliminare dalla memoria il generatore del SUPERTURBO. Scrivete quindi un programma, oppure caricatene uno in memoria. Per "turbosavare" questo programma su nastro, digitate:

TURBOSAVE "nome file"
e premete RETURN. (Si può abbreviare

la parola chiave TURBOSAVE in TSAVE). Verrà visualizzato il solito messaggio:

PRESS RECORD & PLAY

Premete dunque questi tasti e preparatevi a stupirvi. Un header (testata) contenente una speciale routine di "TurboLoad" viene scritta sul nastro (nel C64 lo schermo scompare mentre viene scritta questa routine). Quindi, una serie di colori apparirà sullo schermo durante la registrazione vera e propria del programma. Alla fine dell'operazione lo schermo ritornerà normale. Se volete verificare il programma "Turbosavato", riavvolgete il nastro e digitate:

TURBOVERIFY "nome file"

(TURBOVERIFY può essere abbreviato in TVERIFY). Vedrete il solito messaggio:

PRESS PLAY

Premete il tasto Play sul registratore. (Nel C64, lo schermo scomparirà durante la verifica del programma). Se doveste ottenere un "?OUT OF MEMORY ERROR", digitate semplicemente TVERIFY senza farlo seguire dal nome del file. Non appena verrà letto l'header riceverete sullo schermo i soliti messaggi; come sempre, è possibile nel C64 eliminare la pausa premendo il tasto Commodore. Se viene trovato qualche errore durante il TURBOVERIFY, lo schermo ritornerà normale con visualizzazione del messaggio "?VERIFY ERROR". Se invece il programma supera la verifica, verrà visualizzato il consueto "VERIFY OK".

Caricamento alla velocità della luce

Non vi serve il SUPERTURBO per caricare programmi che siano stati "Turbosavati". Digitate, come normalmente, LOAD e tutto avverrà come al solito, con la differenza che il caricamento avverrà a velocità "Turbo". Ecco ora alcune note aggiuntive sul SUPERTURBO.

Per salvare programmi in linguaggio macchina occorre specificarne gli indirizzi iniziale e finale. Ad esempio, se il vostro programma in linguaggio macchina risiede in memoria tra gli indirizzi 864 e 890, salvatelo in questo modo:

TSAVE "nome file",864,891

Si noti che occorre usare l'indirizzo finale più uno. Per salvare l'intero contenuto della RAM del C64, compresa quella "sotto" la ROM, digitate:

TSAVE "nome",2049,52606

Come detto, molte utility sono perfetta-

mente compatibili con il SUPERTURBO. Se usate il SIMON'S BASIC, tuttavia, dovete evitare il RUN/STOP + RESTORE, ed il comando COLD non ha effetto. Se inoltre usate il DOS WEDGE, scegliete l'opzione 1 per rilocare il SUPERTURBO nella RAM del BASIC ed evitare conflitti tra i vari programmi in linguaggio macchina.

A causa dell'alta velocità sarà opportuno usare solo nastri di alta qualità per il proprio archivio (a proposito, usate comunque sempre cassette con regolazione BIAS normale ed equalizzazione a 120 microsecondi - cioè cassette normali - in quanto l'utilizzo di cassette al cromo, metal e così via, non essendovi predisposto il registratore, provoca notevoli inconvenienti nel caricamento). Il SUPERTURBO, come del resto avviene comunemente anche senza di esso - e lo sappiamo bene noi di SUPERCOMMODORE...! - può, a causa delle differenti tarature delle testine nei registratori, non essere in grado di caricare programmi salvati da altre unità a cassette. Se si ha motivo di supporre che le testine del proprio registratore siano fuori allineamento (tipicamente quando, malgrado un'accurata pulizia delle stesse, non si riescano a caricare programmi registrati da terzi) e non si desideri rivolgersi ad un tecnico per la taratura del registratore, si può agire sulla vitina di regolazione dell'azimut delle testine, posta in corrispondenza del forellino sulla parte superiore dello chas-

sis del Datasette. Tale vitina, che compare sotto il foro quando si preme il tasto Play, dovrà essere ruotata per successivi intervalli seguiti da prove di caricamento, per un massimo di mezzo giro. Si provi prima in senso orario: se l'esito è sempre negativo, la si riporti nella posizione originaria e si ripeta la procedura in senso antiorario, fino a che le prove di caricamento diano esito positivo.

Ritornando al SUPERTURBO, si noti che i programmi "Turbosalvati" dal C64 non possono essere caricati sul VIC 20, e viceversa. Ciò a causa delle differenze tra le due versioni del SUPERTURBO, che verranno analizzate sul prossimo numero.

Si possono usare i comandi LOAD, TSAVE e TVERIFY solo in modo diretto, e non da programma. Inoltre, il SUPERTURBO non può cercare su una cassetta contenente vari programmi "Turbosalvati" un file contrassegnato da un certo nome: occorrerà invece oltrepassare, premendo F.FWD, i programmi che non interessa caricare. A questo proposito è molto utile utilizzare il contametri del nastro. Nessun programma "Turbosalvato" si rilocherà nel caricamento: esso verrà sempre e comunque caricato nella stessa area di memoria dalla quale è stato salvato. In pratica, è come se si usasse sempre il comando LOAD",1,1.

Ciò è particolarmente importante se usate il VIC 20 con diverse espansioni

di memoria: un programma "Turbosalvato" da un VIC espanso non caricherà normalmente in un VIC inespanso, e viceversa.

Quanto è veloce?

I nostri test hanno appurato che un programma da 12 Kbyte impiega 34 secondi per essere caricato da un disk drive 1541, e 44 secondi per essere caricato da nastro usando SUPERTURBO. Tuttavia, il tempo effettivo di caricamento con il SUPERTURBO è di solo 28 secondi, se si prescinde dal tempo speso per leggere l'header in plastica all'inizio del nastro: abbiamo infatti eseguito questi test utilizzando una cassetta completamente riavvolta, presumendo che molte persone non posizionino il nastro esattamente sul punto di inizio di un programma, ma un po' prima. Se invece si posiziona il nastro esattamente in corrispondenza dell'inizio di un programma, con il SUPERTURBO esso viene caricato con una velocità superiore a quella del disk drive.

Il Tsave di un programma da 12 Kbyte impiega 42 secondi, contro i 40 del disk drive 1541.

SUPERTURBO è una delle utility più interessanti e... fondamentali che possa avere chi non dispone di un disk drive!

A proposito, da questo numero in poi, tutte le cassette allegate a SUPERCOMMODORE contengono i programmi registrati in formato SUPERTURBO!!!

Generatore Superturbo versione per VIC 20

```

10 FIS="TURBO/VIC":D=1:REM PER DISCO CAMB
   IA 1 IN 8 :rem 50
20 PRINT"{CLR}{ 4 GIU' } { 2 SPAZI } MEMORIZZ
   .SUPERTURBO { 5 GIU' } " :rem 37
30 FOR I=4097 TO 5260 :rem 20
40 READ A:POKE I,A:CK=CK+A:NEXT :rem 88
50 IF CK>123374 THEN PRINT"{RVS}ERRORE NEL
   LE ISTRUZIONI DATA!" :STOP :rem 92
60 PRINTTAB(7)"{RVS}DATA OK { 2 GIU' }":PRI
   NT"PREMI UN TASTO PER SALVARE..."
   :rem 179
70 GET A$:IF A$="" THEN 70 :rem 241
80 PRINT"{CLR}POKE43,1:POKE44,16:POKE45,1
   41:POKE46,20 { 2 GIU' } " :rem 58
90 PRINT"SAVE"CHR$(34);FIS;CHR$(34);":D
   :rem 243
100 POKE631,19:POKE632,13:POKE633,13:POKE
   198,3:END :rem 148
4097 DATA 32,16,100,0,153,34 :rem 87
4103 DATA 147,32,32,32,144,18 :rem 139
4109 DATA 32,84,85,82,66,79 :rem 72
4115 DATA 83,65,86,69,47,76 :rem 76
4121 DATA 79,65,68,32,31,17 :rem 56
4127 DATA 0,61,16,110,0,158 :rem 34
4133 DATA 40,194,40,52,53,41 :rem 91
4139 DATA 170,50,53,54,172,194 :rem 205
4145 DATA 40,52,54,41,171,55 :rem 96
4151 DATA 57,53,41,58,156,0 :rem 51
4157 DATA 112,16,120,0,153,34 :rem 133
4163 DATA 78,79,87,32,65,84 :rem 78
4169 DATA 34,194,40,49,57,51 :rem 114

```

```

4175 DATA 41,170,50,53,54,172 :rem 148
4181 DATA 194,40,49,57,52,41 :rem 107
4187 DATA 34,45,34,194,40,49 :rem 112
4193 DATA 55,52,41,170,50,53 :rem 98
4199 DATA 54,172,194,40,49,55 :rem 169
4205 DATA 53,41,0,155,16,130 :rem 84
4211 DATA 0,78,36,178,199,40 :rem 109
4217 DATA 51,52,41,170,34,78 :rem 100
4223 DATA 65,77,69,34,170,199 :rem 171
4229 DATA 40,51,52,41,58,79 :rem 58
4235 DATA 80,36,178,78,36,170 :rem 165
4241 DATA 34,91,44,66,65,44 :rem 57
4247 DATA 69,65,93,0,229,16 :rem 65
4253 DATA 140,0,90,83,36,178 :rem 102
4259 DATA 34,83,89,83,40,54 :rem 69
4265 DATA 56,48,41,34,170,79 :rem 114
4271 DATA 80,36,58,90,86,36 :rem 66
4277 DATA 178,34,83,89,83,40 :rem 124
4283 DATA 54,56,51,41,34,170 :rem 101
4289 DATA 78,36,58,84,83,36 :rem 82
4295 DATA 178,34,84,83,65,86 :rem 129
4301 DATA 69,34,170,79,80,36 :rem 109
4307 DATA 58,84,86,36,178,34 :rem 123
4313 DATA 84,86,69,82,73,70 :rem 69
4319 DATA 89,34,170,78,36,0 :rem 63
4325 DATA 11,17,150,0,153,34 :rem 84
4331 DATA 17,144,18,84,79,32 :rem 108
4337 DATA 83,65,86,69,58,31 :rem 75
4343 DATA 17,34,58,153,84,83 :rem 112
4349 DATA 36,44,34,32,32,79 :rem 60
4355 DATA 82,34,58,153,90,83 :rem 114
4361 DATA 36,0,50,17,160,0 :rem 241
4367 DATA 153,34,144,18,84,79 :rem 168
4373 DATA 32,86,69,82,73,70 :rem 68
4379 DATA 89,58,31,17,34,58 :rem 75

```

LOAD**Superturbo**

```

4385 DATA 153,84,86,36,44,34 :rem 117
4391 DATA 32,32,79,82,34,58 :rem 63
4397 DATA 153,90,86,36,0,112 :rem 106
4403 DATA 17,170,0,153,34,17 :rem 89
4409 DATA 144,18,84,79,32,76 :rem 119
4415 DATA 79,65,68,58,31,17 :rem 70
4421 DATA 34,58,153,34,76,79 :rem 114
4427 DATA 65,68,32,79,82,32 :rem 68
4433 DATA 76,79,65,68,34,78 :rem 80
4439 DATA 36,44,34,32,40,84 :rem 55
4445 DATA 85,82,66,79,32,78 :rem 78
4451 DATA 79,84,32,82,69,81 :rem 71
4457 DATA 85,73,82,69,68,41 :rem 77
4463 DATA 0,0,0,56,165,45 :rem 199
4469 DATA 133,90,233,127,133,95 :rem 4
4475 DATA 165,46,133,91,233,2 :rem 155
4481 DATA 133,96,165,55,133,88 :rem 218
4487 DATA 133,174,166,56,134,89 :rem 17
4493 DATA 134,175,32,191,195,230 :rem 51
4499 DATA 89,165,88,166,89,133 :rem 242
4505 DATA 193,134,194,172,0,3 :rem 148
4511 DATA 140,166,2,172,1,3 :rem 35
4517 DATA 140,167,2,141,0,3 :rem 37
4523 DATA 142,1,3,133,55,134 :rem 88
4529 DATA 56,162,255,32,210,251 :rem 250
4535 DATA 32,27,253,32,27,253 :rem 151
4541 DATA 32,17,253,176,75,160 :rem 204
4547 DATA 0,177,172,201,190,208 :rem 250
4553 DATA 240,232,169,32,224,6 :rem 199
4559 DATA 240,13,224,7,208,2 :rem 98
4565 DATA 169,157,224,11,208,5 :rem 208
4571 DATA 169,189,46,169,162,145 :rem 70
4577 DATA 172,200,24,177,172,101 :rem 42
4583 DATA 193,145,172,8,200,17 :rem 4
4589 DATA 172,201,160,240,249,40 :rem 45
4595 DATA 101,194,145,172,224,4 :rem 253
4601 DATA 176,188,157,170,2,232 :rem 254
4607 DATA 136,177,172,157,168,2 :rem 8
4613 DATA 232,169,76,157,166,2 :rem 212
4619 DATA 208,170,96,32,121,0 :rem 148
4625 DATA 168,32,115,0,192,148 :rem 203
4631 DATA 208,8,190,102,0,208 :rem 141
4637 DATA 10,108,166,2,192,149 :rem 205
4643 DATA 208,249,190,29,0,76 :rem 162
4649 DATA 116,196,169,0,133,10 :rem 205
4655 DATA 32,209,225,169,0,32 :rem 152
4661 DATA 213,255,176,56,162,209 :rem 54
4667 DATA 142,161,3,202,142,163 :rem 244
4673 DATA 3,169,96,141,206,3 :rem 109
4679 DATA 32,81,3,8,169,145 :rem 67
4685 DATA 32,210,255,32,112,246 :rem 246
4691 DATA 40,208,8,32,17,253 :rem 103
4697 DATA 144,3,76,141,225,162 :rem 209
4703 DATA 28,76,55,196,32,11 :rem 108
4709 DATA 226,32,138,205,32,247 :rem 254
4715 DATA 215,165,20,166,21,96 :rem 204
4721 DATA 76,246,224,169,183,190 :rem 59
4727 DATA 202,160,1,32,189,255 :rem 201
4733 DATA 169,1,170,168,32,186 :rem 211
4739 DATA 255,134,171,32,121,0 :rem 195
4745 DATA 201,34,208,32,136,230 :rem 242
4751 DATA 122,177,122,240,4,201 :rem 237
4757 DATA 34,208,8,198,171,165 :rem 221
4763 DATA 171,208,240,169,32,190 :rem 50
4769 DATA 201,1,232,224,17,144 :rem 196
4775 DATA 242,32,121,0,240,90 :rem 141
4781 DATA 169,34,32,255,206,240 :rem 0
4787 DATA 83,190,85,0,133,78 :rem 120
4793 DATA 134,79,190,85,0,133 :rem 163
4799 DATA 80,134,81,190,12,1 :rem 106
4805 DATA 176,230,162,13,190,235 :rem 43
4811 DATA 0,157,44,3,202,16 :rem 39
4817 DATA 247,162,44,160,3,134 :rem 202
4823 DATA 193,132,194,162,60,134 :rem 47
4829 DATA 174,132,175,70,157,32 :rem 5
4835 DATA 133,246,8,32,138,255 :rem 210
4841 DATA 6,157,40,176,131,169 :rem 209
4847 DATA 16,133,183,32,44,247 :rem 209
4853 DATA 190,21,1,96,46,3 :rem 4
4859 DATA 173,28,145,133,139,9 :rem 220
4865 DATA 12,141,28,145,208,23 :rem 201
4871 DATA 160,3,185,43,0,153 :rem 98
4877 DATA 78,0,136,16,247,48 :rem 121
4883 DATA 179,230,78,208,2,230 :rem 211
4889 DATA 79,165,78,197,80,165 :rem 243
4895 DATA 79,229,81,96,88,162 :rem 190
4901 DATA 0,160,0,132,192,173 :rem 136
4907 DATA 28,145,133,139,9,12 :rem 158
4913 DATA 141,28,145,202,208,253 :rem 41
4919 DATA 136,208,250,120,132,83 :rem 44
4925 DATA 169,2,190,138,1,198 :rem 165
4931 DATA 83,208,247,169,9,133 :rem 217
4937 DATA 83,165,83,190,138,1 :rem 166
4943 DATA 198,83,208,247,165,78 :rem 25
4949 DATA 190,138,1,165,79,190 :rem 221
4955 DATA 141,1,165,80,190,141 :rem 200
4961 DATA 1,165,81,190,141,1 :rem 97
4967 DATA 162,4,190,134,1,132 :rem 150
4973 DATA 171,177,78,190,146,1 :rem 218
4979 DATA 190,6,1,144,246,234 :rem 162
4985 DATA 165,171,190,146,1,190 :rem 5
4991 DATA 141,1,140,160,2,32 :rem 87
4997 DATA 207,252,165,139,141,28 :rem 62
5003 DATA 145,133,192,169,27,141 :rem 42
5009 DATA 15,144,96,202,208,253 :rem 250
5015 DATA 96,162,3,44,162,5 :rem 50
5021 DATA 190,134,1,133,189,24 :rem 192
5027 DATA 101,171,133,171,169,8 :rem 247
5033 DATA 133,164,190,190,1,166 :rem 245
5039 DATA 163,202,208,253,190,190 :rem 93
5045 DATA 1,70,189,162,13,144 :rem 148
5051 DATA 2,162,41,134,163,198 :rem 197
5057 DATA 164,240,19,162,9,190 :rem 206
5063 DATA 134,1,234,240,223,173 :rem 238
5069 DATA 32,145,73,8,141,32 :rem 102
5075 DATA 145,141,15,144,96,32 :rem 202
5081 DATA 32,32,32,32,32,32 :rem 34
5087 DATA 32,32,32,32,32,32 :rem 40
5093 DATA 32,32,32,32,138,255 :rem 147
5099 DATA 200,132,192,202,208,253 :rem 89
5105 DATA 136,208,250,120,173,27 :rem 35
5111 DATA 145,41,63,141,27,145 :rem 191
5117 DATA 169,248,141,22,145,32 :rem 251
5123 DATA 225,3,102,189,169,2 :rem 148
5129 DATA 197,189,208,245,133,123 :rem 108
5135 DATA 160,9,32,209,3,201 :rem 90
5141 DATA 2,240,249,196,189,208 :rem 4
5147 DATA 230,32,209,3,136,208 :rem 195
5153 DATA 246,133,172,32,209,3 :rem 196
5159 DATA 133,173,32,209,3,133 :rem 198
5165 DATA 174,32,209,3,133,175 :rem 202
5171 DATA 132,171,32,209,3,145 :rem 192
5177 DATA 172,36,16,24,101,171 :rem 197

```



```

5183 DATA 133,171,32,27,253,32 :rem 196
5189 DATA 17,253,144,236,32,209 :rem 3
5195 DATA 3,140,160,2,32,207 :rem 89
5201 DATA 252,165,139,141,28,145:rem 42
5207 DATA 88,133,192,32,66,246 :rem 212
5213 DATA 134,45,132,46,165,189 :rem 0
5219 DATA 197,171,76,154,225,169:rem 65
5225 DATA 8,133,163,32,225,3 :rem 94
5231 DATA 102,189,198,163,208,247 :rem 105
5237 DATA 165,189,96,169,2,44 :rem 174
5243 DATA 45,145,240,251,173,29 :rem 251
5249 DATA 145,72,173,33,145,142 :rem 255
5255 DATA 21,145,104,10,10,96 :rem 138

```

```

2241 DATA 58,137,49,52,48,0 :rem 55
2247 DATA 216,8,130,0,139,65 :rem 98
2253 DATA 36,179,177,34,50,34 :rem 158
2259 DATA 167,49,50,48,0,245 :rem 111
2265 DATA 8,140,0,158,40,194 :rem 98
2271 DATA 40,52,53,41,170,50 :rem 87
2277 DATA 53,54,172,194,40,52 :rem 156
2283 DATA 54,41,171,56,49,50 :rem 105
2289 DATA 41,58,156,0,62,9 :rem 10
2295 DATA 150,0,153,34,147,17 :rem 146
2301 DATA 17,17,32,32,32,32 :rem 32
2307 DATA 78,79,87,32,76,79 :rem 82
2313 DATA 67,65,84,69,68,32 :rem 69
2319 DATA 65,84,58,34,194,40 :rem 114
2325 DATA 49,57,51,41,170,50 :rem 99
2331 DATA 53,54,172,194,40,49 :rem 153
2337 DATA 57,52,41,34,45,34 :rem 52
2343 DATA 194,40,49,55,52,41 :rem 103
2349 DATA 170,50,53,54,172,194 :rem 206
2355 DATA 40,49,55,53,41,34 :rem 52
2361 DATA 17,17,17,17,0,128 :rem 45
2367 DATA 9,160,0,78,36,178 :rem 64
2373 DATA 199,40,51,52,41,170 :rem 150
2379 DATA 34,78,65,77,69,34 :rem 80
2385 DATA 170,199,40,51,52,41 :rem 153
2391 DATA 58,80,36,178,78,36 :rem 123
2397 DATA 170,34,91,44,66,65 :rem 115
2403 DATA 44,69,65,93,34,58 :rem 65
2409 DATA 90,83,36,178,34,32 :rem 110
2415 DATA 32,79,82,32,83,89 :rem 66
2421 DATA 83,40,54,56,48,41 :rem 51
2427 DATA 34,170,80,36,0,195 :rem 100
2433 DATA 9,170,0,90,86,36 :rem 3
2439 DATA 178,34,32,32,79,82 :rem 115
2445 DATA 32,83,89,83,40,54 :rem 62
2451 DATA 56,51,41,34,170,78 :rem 102
2457 DATA 36,58,84,83,36,178 :rem 126
2463 DATA 34,84,85,82,66,79 :rem 75
2469 DATA 83,65,86,69,34,170 :rem 125
2475 DATA 80,36,58,84,86,36 :rem 73
2481 DATA 178,34,84,85,82,66 :rem 123
2487 DATA 79,86,69,82,73,70 :rem 83
2493 DATA 89,34,170,78,36,0 :rem 64
2499 DATA 239,9,180,0,153,34 :rem 110
2505 DATA 32,32,84,79,32,83 :rem 56
2511 DATA 65,86,69,58,32,34 :rem 64
2517 DATA 84,83,36,58,153,34 :rem 114
2523 DATA 32,32,163,163,163,163 :rem 244
2529 DATA 163,163,163,34,58,153 :rem 3
2535 DATA 44,90,83,36,34,17 :rem 57
2541 DATA 17,0,31,10,190,0 :rem 233
2547 DATA 153,34,32,32,84,79 :rem 110
2553 DATA 32,86,69,82,73,70 :rem 66
2559 DATA 89,58,32,34,84,86 :rem 79
2565 DATA 36,58,153,34,32,32 :rem 104
2571 DATA 163,163,163,163,163,163 :rem 97

```

```

2577 DATA 163,163,163,34,58,153 :rem 6
2583 DATA 44,90,86,36,34,17 :rem 63
2589 DATA 17,0,88,10,200,0 :rem 249
2595 DATA 153,34,32,32,84,79 :rem 113
2601 DATA 32,76,79,65,68,58 :rem 71
2607 DATA 32,76,79,65,68,32 :rem 69
2613 DATA 40,84,85,82,66,79 :rem 69
2619 DATA 32,78,79,84,32,78 :rem 76
2625 DATA 69,67,69,83,83,65 :rem 81
2631 DATA 82,89,41,34,58,153 :rem 111
2637 DATA 34,32,32,163,163,163 :rem 199
2643 DATA 163,163,163,163,0,0 :rem 141
2649 DATA 0,56,165,45,133,90 :rem 107
2655 DATA 233,130,133,95,165,46 :rem 255
2661 DATA 133,91,233,2,133,96 :rem 150

```

Generatore Superturbo versione per C64

```

10 FI$="TURBO/64":D=1:REM CAMBIA 1 IN 8 P
ER DISCO :rem 186
20 PRINT"{CLR}{ 3 GIU' }"TAB(11)"MEMORIZZ.
SUPERTURBO( 3 GIU' )" :rem 192
30 FORI=2049 TO 3461 :rem 16
40 READA:POKEI,A:CK=CK+A:NEXT :rem 88
50 IF CK<>143099 THENPRINT"{ 2 SPAZI }
{RVS}ERRORE NELLE ISTRUZIONI DATA!":ST
OP :rem 98
60 PRINTTAB(16)"{RVS}DATA OK{ 3 GIU' }":PR
INTTAB(4)"PREMI UN TASTO PER SALVARE..
." :rem 80
70 GETA$:IFA$=""THEN70 :rem 241
80 PRINT"{CLR}POKE43,1:POKE44,8:POKE45,13
4:POKE46,13{ 2 GIU' }" :rem 15
90 PRINT"SAVE"CHR$(34);FI$;CHR$(34);";":D
:rem 46
100 POKE631,19:POKE632,13:POKE633,13:POKE
198,3:END :rem 148
2049 DATA 62,8,100,0,153,34 :rem 38
2055 DATA 147,17,17,17,34,163 :rem 151
2061 DATA 49,50,41,34,18,32 :rem 43
2067 DATA 84,85,82,66,79,83 :rem 79
2073 DATA 65,86,69,47,76,79 :rem 82
2079 DATA 65,68,32,146,17,17 :rem 113
2085 DATA 17,17,34,58,153,34 :rem 105
2091 DATA 32,32,32,32,82,69 :rem 47
2097 DATA 76,79,67,65,84,69 :rem 88
2103 DATA 32,84,79,58,17,17 :rem 58
2109 DATA 0,140,8,110,0,153 :rem 26
2115 DATA 34,32,32,32,32,32 :rem 31
2121 DATA 32,32,49,46,32,66 :rem 46
2127 DATA 65,83,73,67,32,82 :rem 62
2133 DATA 65,77,32,40,80,79 :rem 57
2139 DATA 73,78,84,69,82,32 :rem 72
2145 DATA 53,53,47,53,54,41 :rem 51
2151 DATA 17,34,58,153,34,32 :rem 96
2157 DATA 32,32,32,32,32,32 :rem 35
2163 DATA 50,46,32,70,82,79 :rem 55
2169 DATA 77,32,53,50,54,48 :rem 61
2175 DATA 54,32,84,48,32,53 :rem 56
2181 DATA 51,50,52,56,17,17 :rem 47
2187 DATA 0,199,8,120,0,151 :rem 45
2193 DATA 49,55,49,44,48,58 :rem 74
2199 DATA 133,34,32,32,32,32 :rem 93
2205 DATA 89,79,85,82,32,67 :rem 73
2211 DATA 72,79,73,67,69,32 :rem 64
2217 DATA 40,49,47,50,41,34 :rem 47
2223 DATA 59,65,36,58,139,65 :rem 118
2229 DATA 36,178,34,49,34,167 :rem 167
2235 DATA 151,49,55,49,44,49 :rem 114

```

LOAD**Superturbo**

```

2667 DATA 165,171,208,6,169,0 :rem 160
2673 DATA 162,208,208,4,165,55 :rem 207
2679 DATA 166,56,133,88,133,174 :rem 16
2685 DATA 134,89,134,175,32,191 :rem 9
2691 DATA 163,230,89,165,88,166 :rem 17
2697 DATA 89,133,193,134,194,172 :rem 67
2703 DATA 0,3,140,166,2,172 :rem 35
2709 DATA 1,3,140,167,2,141 :rem 39
2715 DATA 0,3,142,1,3,164 :rem 190
2721 DATA 171,240,4,133,55,134 :rem 190
2727 DATA 56,162,255,32,142,251 :rem 252
2733 DATA 32,219,252,32,219,252 :rem 249
2739 DATA 32,209,252,176,75,160 :rem 5
2745 DATA 0,177,172,201,190,208 :rem 248
2751 DATA 240,232,169,32,224,6 :rem 197
2757 DATA 240,13,224,7,208,2 :rem 96
2763 DATA 169,157,224,11,208,5 :rem 206
2769 DATA 169,189,44,169,162,145 :rem 75
2775 DATA 172,200,24,177,172,101 :rem 40
2781 DATA 193,145,172,8,200,177 :rem 2
2787 DATA 172,201,160,240,249,40 :rem 43
2793 DATA 101,194,145,172,224,4 :rem 251
2799 DATA 176,188,157,170,2,232 :rem 14
2805 DATA 136,177,172,157,168,2 :rem 6
2811 DATA 232,169,76,157,166,2 :rem 210
2817 DATA 208,170,96,32,121,0 :rem 146
2823 DATA 168,32,115,0,192,148 :rem 201
2829 DATA 208,8,190,104,0,208 :rem 150
2835 DATA 10,108,166,2,192,149 :rem 203
2841 DATA 208,249,190,31,0,162 :rem 197
2847 DATA 128,108,166,2,169,0 :rem 158
2853 DATA 133,10,32,212,225,169 :rem 243
2859 DATA 0,32,213,255,176,56 :rem 158
2865 DATA 162,209,142,165,3,202 :rem 249
2871 DATA 142,167,3,169,96,141 :rem 213
2877 DATA 209,3,32,81,3,8 :rem 213
2883 DATA 169,145,32,210,255,32 :rem 254
2889 DATA 216,245,40,208,8,32 :rem 160
2895 DATA 209,252,144,3,76,141 :rem 209
2901 DATA 225,162,28,76,55,164 :rem 208
2907 DATA 32,14,226,32,138,173 :rem 200
2913 DATA 32,247,183,165,20,166 :rem 254
2919 DATA 21,96,76,249,224,169 :rem 225
2925 DATA 188,190,200,160,1,32 :rem 194
2931 DATA 189,255,169,1,170,168 :rem 11
2937 DATA 32,186,255,134,171,32 :rem 1
2943 DATA 121,0,201,34,208,32 :rem 133
2949 DATA 136,230,122,177,122,240 :rem 92
2955 DATA 4,201,34,208,8,198 :rem 109
2961 DATA 171,165,171,208,240,169 :rem 102
2967 DATA 32,190,199,1,232,224 :rem 208
2973 DATA 17,144,242,32,121,0 :rem 141
2979 DATA 240,86,169,34,32,255 :rem 221
2985 DATA 174,240,79,190,87,0 :rem 169
2991 DATA 133,78,134,79,190,87 :rem 226
2997 DATA 0,133,80,134,81,190 :rem 155
3003 DATA 10,1,176,230,162,9 :rem 83
3009 DATA 190,237,0,157,44,3 :rem 96
3015 DATA 202,16,247,162,44,160 :rem 239
3021 DATA 3,134,193,132,194,162 :rem 241
3027 DATA 60,134,174,132,175,70 :rem 246

```

```

3033 DATA 157,32,237,245,8,32 :rem 149
3039 DATA 21,253,6,157,40,176 :rem 151
3045 DATA 131,169,16,133,183,32 :rem 246
3051 DATA 147,246,190,19,1,96 :rem 155
3057 DATA 46,3,165,1,9,48 :rem 212
3063 DATA 133,139,208,27,160,3 :rem 195
3069 DATA 185,43,0,153,78,0 :rem 53
3075 DATA 136,16,247,48,183,230 :rem 0
3081 DATA 78,208,2,230,79,165 :rem 158
3087 DATA 78,197,80,165,79,229 :rem 233
3093 DATA 81,96,88,162,0,160 :rem 109
3099 DATA 0,132,192,173,17,208 :rem 202
3105 DATA 41,239,141,17,208,165 :rem 246
3111 DATA 1,133,139,41,254,133 :rem 184
3117 DATA 1,202,208,253,136,208 :rem 239
3123 DATA 250,120,132,83,169,2 :rem 188
3129 DATA 190,138,1,198,83,208 :rem 211
3135 DATA 247,169,9,133,83,165 :rem 214
3141 DATA 83,190,138,1,198,83 :rem 158
3147 DATA 208,247,165,78,190,138 :rem 61
3153 DATA 1,165,79,190,141,1 :rem 96
3159 DATA 165,80,190,141,1,165 :rem 201
3165 DATA 81,190,141,1,162,4 :rem 92
3171 DATA 190,134,1,132,171,177 :rem 243
3177 DATA 78,190,146,1,190,4 :rem 107
3183 DATA 1,144,246,234,165,171 :rem 249
3189 DATA 190,146,1,190,141,1 :rem 146
3195 DATA 140,160,2,32,147,252 :rem 192
3201 DATA 165,139,9,48,133,1 :rem 98
3207 DATA 133,192,96,202,208,253 :rem 44
3213 DATA 96,162,3,44,162,5 :rem 48
3219 DATA 190,134,1,133,189,24 :rem 199
3225 DATA 101,171,133,171,169,8 :rem 245
3231 DATA 133,164,190,190,1,166 :rem 243
3237 DATA 163,202,208,253,190,190 :rem 91
3243 DATA 1,70,189,162,13,144 :rem 146
3249 DATA 2,162,41,134,163,198 :rem 204
3255 DATA 164,240,17,162,9,190 :rem 202
3261 DATA 134,1,234,240,223,165 :rem 237
3267 DATA 1,73,8,133,1,238 :rem 0
3273 DATA 32,208,96,32,32,32 :rem 98
3279 DATA 32,32,32,32,32,32 :rem 41
3285 DATA 32,32,32,32,32,32 :rem 38
3291 DATA 32,173,17,208,41,239 :rem 202
3297 DATA 141,17,208,32,21,253 :rem 197
3303 DATA 200,132,192,169,54,133 :rem 35
3309 DATA 1,202,208,253,136,208 :rem 242
3315 DATA 250,120,169,248,141,6 :rem 246
3321 DATA 221,32,228,3,102,189 :rem 189
3327 DATA 169,2,197,189,208,245 :rem 15
3333 DATA 133,123,160,9,32,212 :rem 185
3339 DATA 3,201,2,240,249,196 :rem 149
3345 DATA 189,208,230,32,212,3 :rem 195
3351 DATA 136,208,246,133,172,32 :rem 40
3357 DATA 212,3,133,173,32,212 :rem 188
3363 DATA 3,133,174,32,212,3 :rem 88
3369 DATA 133,175,132,171,32,212 :rem 40
3375 DATA 3,145,172,36,16,24 :rem 101
3381 DATA 101,171,133,171,32,219 :rem 33
3387 DATA 252,32,209,252,144,236 :rem 49
3393 DATA 32,212,3,140,160,2 :rem 83
3399 DATA 32,147,252,165,139,133 :rem 56
3405 DATA 1,88,133,192,32,170 :rem 147
3411 DATA 245,134,45,132,46,165 :rem 247
3417 DATA 189,197,171,76,154,225 :rem 65
3423 DATA 169,8,133,163,32,228 :rem 204
3429 DATA 3,102,189,198,163,208 :rem 6
3435 DATA 247,165,189,96,169,16 :rem 22
3441 DATA 44,13,220,240,251,173 :rem 235
3447 DATA 13,221,142,7,221,72 :rem 141
3453 DATA 169,25,141,15,221,104 :rem 242
3459 DATA 74,74,96 :rem 140

```

Newsletter Editor

di **C.F. Thompson**
trad. e adatt.
di **U. G. Barzagli**

"Newsletter Editor" (NED) era, in origine, un semplice progetto di programmazione per ottenere un risultato molto preciso. Lo scopo era di poter leggere qualsiasi file di testo registrato su disco, scritto mediante un qualsiasi word processor senza utilizzare il programma di trattamento di testi da cui il testo era stato in origine generato.

Il mio compito come vicesegretario del Commodore Computer Club della mia città consiste nello stampare gli articoli mensili (provenienti da autori diversi) sulla mia stampante Dynax DX15, in modo che il segretario disponga delle copie per la composizione definitiva. All'inizio sembrava un compito di poco conto, quindi accettai il lavoro nel pieno delle mie facoltà mentali e con l'ingenuità dei neofiti.

Scoprii ben presto per quale ragione mi era stato chiesto di ricoprire questo ruolo. Pensavo fosse dovuto al fatto di possedere la miglior stampante del club. Errato. Ho ottenuto il lavoro perché possedevo il maggior numero di programmi di trattamento testi.

Ora, ogni mese, ricevo articoli su disco, scritti tramite fino a sei word processor diversi. Non posso chiedere a tutti gli autori di utilizzare il "Quick Brown Fox" solo perché è il mio preferito: quindi, per legittima difesa, ho scritto NED.

Obiettivo del programma

"Newsletter Editor" è stato progettato come strumento per redattori di articoli, ma chiunque abbia bisogno di leggere informazioni contenute in file su disco potrà trovarlo utile. Il programma non consente di correggere testi, ma piuttosto di estrarli da disco, mostrarli sullo schermo e stamparli su carta. Possono essere "catturati" testi scritti con gli otto word processor più diffusi. La figura 1 rappresenta un'uscita su schermo del menu principale per la versione per C64. È ciò che vedrete apparire sullo schermo mandando in esecuzione il programma. NED funzionerà sul C64 e sul



VIC 20 con espansione aggiuntiva da 8K RAM.

Esaminando i vari word processor sul mercato, ho scoperto che tutti registrano i loro file di testo su disco in uno di questi tre formati: SEquenziale, Utente (in inglese USeR, abbreviato in USR) o PRoGramma (PRG).

All'interno di ciascuno di questi tre formati il testo può essere registrato in una qualsiasi delle tre serie di codici: PET ASCII, ASCII o Codice di Schermo. Per produrre un lettore di testi veramente universale tutte le opzioni di registrazione vanno prese in considerazione. Dal momento che non pretendo voi sappiate che i testi ottenuti con QBF vengono registrati come file di programma in codice ASCII a 7 bit o che Word Mañager

produce file d'utente in codice PET ASCII, ecc., ho etichettato le voci del menu principale con gli stessi nomi dei diversi word processor. Ciò vi libererà dal peso della maggior parte del lavoro di interpretazione.

Chi può utilizzare il programma?

NED è stato progettato in base all'ipotesi che voi non possediate tutti i word processor che siano mai stati realizzati. Presumo anche che riceviate del materiale in un formato che non siete in grado di leggere, a meno che l'autore utilizzi lo stesso word processor che utilizzate voi. Si presume anche che disponiate di una stampante ad interfaccia parallela o di una stampante VIC.



Come usare il programma.

Caricate (LOAD) e mandate in esecuzione (RUN) "Newsletter Editor". Il menu principale dispone di tredici voci diverse.

Le prime nove vi consentono di scegliere con che "tipo" di file di testo lavorare. La opzione contraddistinta dal numero 10 vi permetterà di uscire dal programma ed iniziare a listare NED sullo schermo. La 11 cambierà il colore del cursore. Le opzioni 12 e 13 sono disponibili solo per il C64 e non appariranno pertanto per gli utenti di VIC 20. Cambiano i colori della cornice e dello schermo.

Inserite il disco contenente il file di testo da leggere nella periferica contraddistinta dal numero 8 (drive 0) e scegliete una delle opzioni da 1 a 9, in modo che corrisponda al tipo del file. Il "tipo" apparirà nell'angolo superiore sinistro. La scritta "Premi i o u" vi ricorda che potete interrompere, in qualsiasi momento, l'uscita del testo o uscire dall'opzione e ritornare al menu principale.

La scritta "Vuoi stampare il testo? s/n/u" vi offre tre possibilità. Rispondendo "u" si esce dall'opzione e si ritorna al menu principale; "s" invia il testo sia allo schermo che alla stampante; "n" si limiterà a visualizzare il testo sullo schermo.

Se scegliete "s", la domanda successiva vi chiederà di specificare "Stampante CBM s/n". Rispondete "s", se dispone di una stampante VIC 1525 o se l'interfaccia della vostra stampante converte il codice PET ASCII in ASCII o se l'interfaccia emula la 1525. La maggior parte delle stampanti a matrice di punti funzionerà rispondendo "s". Provate sempre ad usare l'opzione "s". Essa visualizzerà il testo nel formato di schermo opportuno. Se il testo visualizzato risulta invertito (il mAIUSCOLO è mINUSCOLO, e viceversa) potete fare due cose per correggere il difetto.

Il metodo migliore richiede un'attenta lettura del manuale della vostra stampante, o dell'interfaccia, per stabilire quale interruttore controlla l'opzione "7 bit o 8 bit". Il cambiamento dello stato di questo interruttore risolverà il problema. L'altro metodo consiste nel rispondere alla domanda con "n". Ciò opererà la conversione, ma il funzionamento non è garantito con qualsiasi stampante. Inoltre il formato risulterà invertito sullo schermo. Cercate di far sì che sia la stampante o l'interfaccia ad operare la conversione di codice al vostro posto. A questo punto il programma richiede il nome del file. Digitatelo esattamente come appare nella "directory" del di-

scio. La sola eccezione è rappresentata dai file di testo prodotti dal word processor Quick Brown Fox. Invertite il formato. Generalmente ciò significa dover inserire il nome del file in formato minuscolo, anche se la "directory" lo mostra in maiuscolo. Se tutto funziona, lo schermo verrà cancellato e vedrete un cursore lampeggiante apparire nell'angolo superiore sinistro dello schermo. Il testo comincerà a riempire lo schermo, un carattere alla volta. Se avete scelto di inviarlo anche alla stampante, comincerà a stampare il testo una riga per volta. L'operazione proseguirà fino a che verrà raggiunta l'etichetta di fine del file (End Of File, EOF), o premiate "u" o "i". Se si incorre in un errore dovuto alla registrazione su disco, lo stato dell'unità dischi apparirà sullo schermo, comunicandovi quale sia il problema, e l'esecuzione del programma verrà interrotta.

Testo formattato?

Molti word processor incorporano dei codici di formattazione all'interno dei rispettivi testi. Non volevo proprio che questi codici apparissero sullo schermo o sull'uscita della stampante, quindi NED "pulirà" alcuni "tipi" di file da questi codici indesiderati. In particolare, QBF e TOTL.TEXT usano parecchi codici di formattazione, ma nessuno di essi verrà visualizzato. Tutto ciò verrà dettagliatamente esaminato in seguito. Potete controllare alcuni aspetti del formato di stampa. Potete fissare il margine destro. Se il formato del vostro articolo è a 40 colonne, ad esempio, potete stampare il testo con un margine destro in 40. Le parole non verranno spezzate. Solo parole intere verranno stampate su ciascuna riga. Non viene effettuata alcuna giustificazione del margine destro, ma può essere implementata, dal momento che la maggior parte del codice necessario è già disponibile.

Potrebbe essere un'ottima idea fornire ai vostri autori una copia di NED, in modo che possano leggere con esso i loro file di testo e vederne l'aspetto prima di presentarveli. In generale, i file di testo andrebbero scritti senza codici di formattazione incorporati. È meglio che l'unico carattere non di testo incorporato sia rappresentato da un RETURN. Approfondiremo l'argomento in seguito.

Note programmatiche e descrizione

NED è scritto in linguaggio BASIC. Uno sguardo al listato vi permetterà di vedere che il programma è composto da 9 moduli principali. Dal momento che esistono parecchie combinazioni possibili di tipi di file, i sottoprogrammi di conversione vengono gestiti in uno dei quattro moduli seguenti: QBF, Word Manager (WM), SpeedScript e Lettore di file Sequenziali.

Ciascuno di questi moduli svolge compiti simili, ma lavora su file di struttura differente.

Un grafo del programma

Indipendentemente da quale tipo di file di testo debba essere letto, il grafo di flusso del programma è all'incirca lo stesso, con variazioni di minor conto. Dopo che i valori delle variabili controllate dall'utente sono stati inseriti NED compie un salto condizionato al modulo che gestisce il tipo di file in questione. Un file viene aperto in lettura sull'unità dischi, utilizzando il nome del file indicato. Viene controllato lo stato della periferica e, se il nome del file richiesto è stato trovato, il programma prosegue. Se si verifica un problema di qualsiasi tipo con il canale dei dischi, lo stato della periferica viene visualizzato sullo schermo per un paio di secondi e quindi riapparirà il secondo menu, in modo possiate ricominciare dall'inizio.

Se lo stato della periferica è corretto, il testo viene letto utilizzando una istruzione GET#. È vero, questa istruzione è piuttosto lenta, ma permette l'esame dei singoli caratteri. La conversione da Codice di Schermo o ASCII viene operata ed il carattere viene riesaminato (in diversi moduli) per stabilire se costituisce parte di un codice di formattazione indesiderato. Se è così, il codice viene scartato o lo si manipola. I caratteri vengono quindi inviati ad un sottoprogramma, che stabilisce se il testo deve essere solo visualizzato o anche stampato. Se deve essere stampato su di una stampante ASCII ("n"), viene convertito in codice ASCII. Il testo da stampare confluisce quindi in una stringa (PS\$) e conservato fino a che si riceve il carattere CHR\$(13) (RETURN) o la lunghezza della stringa raggiunge il margine destro. Quando la seconda condizione è verificata si esamina da destra a sinistra la variabile PS\$ fino a che viene individuato uno spazio vuoto (32). Quando un carattere avente codice di schermo 32 viene individuato tutto il testo alla sua sinistra viene stampato. La parte di testo alla destra viene memorizzata nella variabile PS\$ ed il processo di costruzione della riga di stampa riprende. In questo modo il testo viene stampato senza spezzare le parole. Potete fissare il margine destro in riga 1 (RM=?). Se si incontra un RETURN prima del completo riempimento di PS\$, il contenuto della variabile viene stampato nel suo formato corrente.

Dopo che un carattere è stato inviato al sottoprogramma di stampa la tastiera viene esaminata per verificare se siano stati premuti i tasti "i" o "u". In caso contrario viene esaminato lo stato della periferica per controllare se è verificata la condizione di fine del file (End Of File, EOF). Se si incontra il segnale corrispondente, o se è stato premuto il tasto "u", il controllo viene trasferito al modulo di gestione dello stato del disco (Disk Status Module), dove lo stato vie-

ne letto dalla periferica e visualizzato sullo schermo. Tutti i file aperti vengono chiusi e ricompare il menu principale.

Particolari

Non esiste un lettore di file realmente "universale". L'obiettivo di NED è limitato ai tipi di word processor più diffusi. Se vi capita di avere a che fare con un word processor che non è elencato nel menu principale, ci sono buone probabilità di recuperare il testo scritto con esso, grazie ad un semplice lavoro investigativo. Innanzi tutto guardate la "directory" e scoprite in che formato il file è stato registrato. NED gestisce i formati PRG, USR e SEQ. Una volta che conoscete il tipo del file, scegliete la voce del menu che opera su quel tipo. Il word processor "Quick Brown Fox" (QBF) produce file PRG in codice ASCII. Word Manager file d'utente (USR) in codice PET ASCII. SpeedScript file PRG in Codice di Schermo. Gli altri word processor utilizzano file SEQUENZIALI in codice PET ASCII. Provate a leggere file di testo creati con altri word processor con le opzioni che gestiscono la struttura dei file. Otterrete effettivamente qualcosa. Esaminiamo ora ciascuno dei quattro moduli principali.

Quick Brown Fox

Il word processor QBF ha bisogno di un breve sottoprogramma che per gli altri word processor risulta superfluo. Quando indicate il nome di un file esso deve venire convertito in codice ASCII vero e proprio dal sottoprogramma che inizia in riga 380. I codici di formattazione incorporati in QBF iniziano tutti con il simbolo #. I caratteri in ingresso vengono esaminati dal sottoprogramma che inizia in riga 270. Se si incontra un simbolo #, NED compie un salto condizionato a riga 470 per cominciare ad esaminare i caratteri seguenti. Alcuni compiti minori del processo di formattazione vengono svolti in queste righe. Se viene trovato un codice #p (nuovo paragrafo), vengono inviati al sottoprogramma di stampa una istruzione RETURN e quattro spazi bianchi, simulando, così, gli effetti del codice. Se si trova #f (=form feed, nuova pagina), le parole "Nuova Pagina" vengono visualizzate sullo schermo per mostrare il cambio di pagina. Tutti gli altri codici vengono estrapolati, in modo che solo il testo venga stampato.

Word Manager

Questo word processor rappresenta l'"anticonformista" del gruppo. WM è il word processor a 80 colonne fornito con la scheda di interfaccia a 80 colonne Data20 (Protecto, ecc.). Il testo viene memorizzato sotto forma di file d'utente (USR) in codice PET ASCII, in formato formattato. Tutto ciò significa che quando leggete un file prodotto da WM esso uscirà sulla stampante esattamente nel formato in cui era stato inserito tramite WM. Nessuna formattazione

viene eseguita da NED. Questo modulo non utilizza il sottoprogramma di stampa di riga 20, ma sfrutta un sottoprogramma personale che inizia in riga 1210. Non potete fissare il margine destro per un file prodotto da WM.

SpeedScript

SpeedScript è comparso per la prima volta lo scorso anno sulle pagine di SUPERCOMMODORE ed ha subito ottenuto un ottimo successo di pubblico. L'ho incluso in questo programma proprio per la sua diffusione. SpeedScript memorizza i suoi testi sotto forma di file PRG in Codice di Schermo. La conversione da Codice di Schermo a PET ASCII viene effettuata alle righe da 2200 a 2270. I primi 48 caratteri del file vengono scartati, vedi riga 2110 in cui la variabile XX è uguale a 48. Ho inserito questo valore in una variabile, in modo che possiate cambiarlo. Se perdetevi parte del testo od ottenete caratteri spuri all'inizio, regolate XX di conseguenza in riga 2050.

File SEQUENZIALI

Qui inizia il divertimento. Preparate dell'Aspirina. La maggior parte dei word processor memorizza i testi sotto forma di file SEQUENZIALI, quindi il modulo di gestione dei file SEQ leggerà QUALCOSA da qualsiasi file SEQ (database, spreadsheet, ecc). I problemi iniziano quando si ha a che fare con qualcosa chiamato "Delimitatore di file" (File Delimiter). Si tratta di un carattere inserito nel testo per limitare la lunghezza della registrazione (generalmente a circa 254 caratteri). Il delimitatore di file più usato è il simbolo " (doppi apici), ma anche CHR\$(13) è molto diffuso. NED estrapolerà questi caratteri di delimitazione indesiderati, ma tutto ciò rappresenta un procedimento piuttosto complicato. Se viene utilizzato CHR\$(13), il simbolo di RETURN all'interno del testo sarà rappresentato da qualche altro carattere, generalmente una freccia o il simbolo " ". La sezione del programma dalla riga 5000 alla 5340 stabilisce quale sia il delimitatore di file ed il simbolo di RETURN per ciascuno dei differenti word processor. La variabile FL ha questo scopo. La riga 3080 controlla la ricorrenza del delimitatore e lo scarta.

TOTL. TEXT

Circa la metà delle istruzioni del modulo SEQ si occupa dei codici di formattazione TT, estrapolandoli o operando su di essi grosso modo come per QBF. I codici di formattazione TT iniziano tutti con il doppio simbolo del dollaro: \$\$, seguito da un codice specifico. Ad esempio, \$\$pr è un comando di "nuovo paragrafo", esattamente come l'istruzione #p di QBF. Se viene trovato questo comando \$\$pr, il programma compie un salto condizionato alla riga 483 del modulo di QBF ed esegue una istruzione simulata di "nuovo paragrafo". Il comando \$.DOL\$\$sk (SKIP a line, salta una riga) viene interpretato come un comando di

"a capo". Tutti gli altri comandi \$\$ vengono estrapolati. La formattazione di TT viene effettuata dalle righe da 3170 a 3290. Il simbolo \$ viene rilevato ed esaminato in riga 3160. Se il carattere successivo è un "\$", si presume che si tratti di un codice di formattazione di TT e si opera su di esso.

Paper Clip

I file di testo di PaperClip possono essere registrati come file SEQUENZIALI mediante l'opzione CTRL Z o come file PRG in Codice di Schermo tramite l'opzione CTRL S. NED si occupa di entrambe, ma preferisce la versione CTRL Z. La versione CTRL S utilizza il modulo di SpeedScript, mentre la CTRL Z il modulo SEQ. Non viene operata alcuna formattazione, dal momento che i codici di formattazione non vengono incorporati di frequente.

WordPro 3

Questo wordprocessor funziona bene con il nostro programma, ma i suoi testi verranno stampati con il margine destro assegnato dal file, quindi se avete bisogno di articoli a quaranta colonne, dichiaratelo nei testi e, in seguito, inizializzate NED in maniera opportuna. La stessa cosa vale per i testi ottenuti con HES, ma fate attenzione a che non vi siano inseriti codici di formattazione incorporati oltre al RETURN, altrimenti essi appariranno sulla vostra stampante.

EasyScript

La maggior parte dei codici incorporati sono concentrati all'inizio del testo, quindi evitate di usare codici di formattazione dopo quelli prefissati all'inizio. Questa precauzione vi permetterà di ottenere una copia priva di errori.

Espedienti programmatici

Ho utilizzato in NED alcune righe di istruzioni BASIC che potreste essere interessati ad incorporare in altri programmi. Le righe da 9040 a 9080 stabiliscono se il programma viene utilizzato da un VIC o da un C64 leggendo il contenuto della locazione 65532 tramite un'istruzione PEEK. Se il risultato è uguale a 34, si sta usando un VIC, mentre se il risultato è 226, si tratta di un C64. Se il risultato è differente, si può trattare sia di un PET (non so di che tipo) o di un VIC/C64 fornito di scheda d'interfaccia da 80 colonne. NED è compatibile con le schede di interfaccia da 80 colonne.

Mentre copiate il programma da queste pagine, potete registrarlo su disco battendo RUN2 e RETURN. Ciò farà sì che la versione precedente di NED venga cancellata dal disco e rimpiazzata dalla nuova. Dal momento che preferisco aver copie aggiuntive dei miei programmi (Back-up), ho inserito lo stesso sottoprogramma in riga 4, con un diverso nome del file. Il comando RUN4, in formato diretto, aggiornerà anche la copia aggiuntiva. Controllate molto attenta-

mente le prime 5 righe in cerca di eventuali errori, dal momento che un errore in una di queste righe può risultare fatale. Il comando RUN6 listerà il programma sulla stampante. Potete indicare la sezione del listato desiderata dopo l'istruzione "LIST" in riga 7. Mentre stavo scrivendo i moduli del programma ho usato l'opzione del menu contraddistinta dal valore 10, per listare automaticamente sullo schermo le righe su cui stavo, di volta in volta, lavorando. La riga 9400 svolge questa funzione. Le

righe da 9120 a 9170 stabiliscono quale calcolatore viene attualmente utilizzato e visualizzano un titolo differente, in base al fatto che si tratti del VIC o del C64.

Annotazioni conclusive

NED è abbondantemente commentato per vostra comodità, ma la maggior parte delle istruzioni di commento (REM) andrebbe rimossa o non dovrebbe essere copiata. Cancellate tutte le

istruzioni REM che condividono il numero di riga con altre istruzioni BASIC. La riga 28 è un esempio di una istruzione REM che andrebbe cancellata. Ciò velocizzerà leggermente il funzionamento del programma. Non cancellate le istruzioni di commento che hanno un proprio numero di riga. La riga 8000 costituisce un esempio di una istruzione REM che dovrete lasciare immutata.

NED non è veloce. Funziona correttamente, ma il testo viene letto all'incirca alla stessa velocità con cui lo leggete.

Newsletter Editor



```

0 REM"NEWSLETTER EDITOR" V1.3.7 :rem 121
1 CC$=CHR$(31):RM=40:GOTO8070 :rem 138
2 OPEN1,8,15,"I0":PRINT#1,"S0:EDITOR"
   :rem 131
3 SAVE"EDITOR",8:GOSUB6000:CLOSE1:END
   :rem 67
4 OPEN1,8,15,"I0":PRINT#1,"S0:EDITOR BU"
   :rem 28
5 SAVE"EDITOR BU",8:GOSUB6000:CLOSE1:END
   :rem 220
6 PRINT"PRINT#4:CLOSE4":CLOSE4 :rem 179
7 OPEN4,4,7:CMD4:LIST :rem 24
8 : :rem 114
20 REM"*****"
   :rem 160
22 REM"*{ 2 SPAZI}SO{ 2 T}OPROGRA{ 3 M}
   A{SPAZI}DI{SPAZI}STAMPA{ 2 SPAZI}*"
   :rem 127
24 REM"*****"
   :rem 164
26 : :rem 162
28 IFP=.THEN50:REM SOLO SCHERMO :rem 180
30 IFP=2THEN38:REM STAMPANTE CBM-VIC
   :rem 247
32 IFC<NETHEN38:REM STAMPANTE ASCII
   :rem 212
33 IFC>191THENC=(C-128):GOTO38 :rem 250
34 IFC>NETHENC=(C-NC):GOTO38 :rem 247
36 IFC>NRTHENC=(C+NC) :rem 24
38 IFC=NATHENI=LEN(PSS):GOTO46 :rem 146
40 PSS=PSS+CHR$(C):IFLEN(PSS)<RMTHEN50
   :rem 30
42 FORI=RMT01STEP-1 :rem 224
44 IFMID$(PSS,I,1)<>" "THENPR$=MID$(PSS,I
   ,1)+PR$:NEXTI :rem 106

```

```

46 PRINT#4,LEFT$(PSS,I):PSS=PR$:PR$="":GO
   TO50 :rem 179
48 PRINT#4,CHR$(C); :rem 76
50 PRINTCHR$(C);:RETURN :rem 220
52 : :rem 161
100 REM"*****"
   :rem 165
110 REM"*{ 6 SPAZI}QUICK{SPAZI}BROWN
   {SPAZI}FOX{ 7 SPAZI}*" :rem 192
120 REM"*****"
   :rem 167
130 : :rem 206
140 OPEN4,4,SA:OPEN8,8,8,NBS+"P,R"
   :rem 208
150 OPEN1,8,15:GOSUB6000:IFDSTHEN370
   :rem 152
170 PRINTCH$; :rem 15
180 GET#8,A$,A$ :rem 246
190 GET#8,A$:GOSUB270:GETI$:IFI$="I"THEN2
   50 :rem 6
200 IFI$="U"THEN240 :rem 41
210 IFST=.THEN190 :rem 253
220 IFST=-128THEN190 :rem 152
240 PRINT#4,PS$:PS$="":GOSUB6000:GOTO9410
   :rem 176
250 GETB$:IFB$=" "THEN250:REM PAUSA
   :rem 235
260 GOTO190 :rem 107
270 C=ASC(A$) :rem 166
280 REMIFC=NATHEN380 :REM RET,NUM
   :rem 11
290 IFC=NDTHENGOSUB470:GOTO350:REM #
   :rem 210
300 IFC>NQTHENC=NA:GOTO350 :rem 139
310 IFC>NFTHENC=C-NC:GOTO350:REM "DA M A
   M" :rem 53
320 IFC<NETHEN350:REM NUM :rem 11
330 IFC<NCTHENC=C+NR:GOTO350 :rem 253
340 C=C+NC:REM "DA M A M" :rem 88
350 GOSUB28 :rem 130
360 NA$="":NB$="":PN$="QUICK BROWN FOX"
   :rem 43
370 GOSUB6080:GOSUB380:GOTO140 :rem 69
380 REM CONVERSIONE FILENAME :rem 11
390 T$="" :rem 149
400 FORN=1TOLEN(NA$):T$=MID$(NA$,N,1)
   :rem 133
410 T=ASC(T$) :rem 198
420 IFT>NFTHENT=T-NC :rem 29
430 IFT>NRTHENT=T+NC :rem 40
440 NB$=NB$+CHR$(T) :rem 14
450 NEXT :rem 216
460 RETURN :rem 122
470 GET#8,A$:C=ASC(A$) :rem 174
480 IFC<>112THEN490 :rem 72
483 C=13:GOSUB28:FORI=1TO4 :rem 86
486 C=32:GOSUB28:NEXT:RETURN:REM #P
   :rem 207

```

```

490 IFC=NLCORC=NKORC=NMTHEN560:REM #NLR
      :rem 179
500 IFC=NOORC=NHTHEN560:REM #D,T:rem 199
510 IFC=NJORC=NGTHEN570:REM #H,C:rem 182
520 IFC=NNORC=NPTHEN570:REM #U,O:rem 221
530 IFC=106THEN570:REM #J      :rem 148
535 IFC=115THENGET#8,AS,AS:C=:RETURN:REM
      #S      :rem 101
540 IFC=102THENPRINTS"* NUOVA{ SPAZI }PAG
      INA *RTS:C=:RETURN      :rem 42
550 RETURN      :rem 122
560 GET#8,AS,AS,AS:C=:RETURN  :rem 139
570 C=:RETURN   :rem 100
580 :           :rem 215
1000 REM"*****"
      :rem 171
1010 REM"*{ 7 SPAZI}WORD{ SPAZI}MANAGER
      { 8 SPAZI}*"      :rem 149
1020 REM"*****"
      :rem 173
1030 :           :rem 254
1040 PMS="WORD MANAGER"      :rem 63
1050 GOSUB6080:OPEN4,4,SA:OPEN1,8,15
      :rem 103
1060 OPEN8,8,8,NA$+" ,U,R":GOSUB6000:IPDST
      HEN1050      :rem 27
1070 PRINTCHS;      :rem 63
1090 FORI=1TO15:GET#8,AS:NEXTI :rem 57
1100 GET#8,AS:C=ASC(AS):IFC=199THEN1160
      :rem 184
1110 IFC>N0THEN1140      :rem 103
1120 GOSUB1210:GETIS:IFIS="I"THEN1170
      :rem 135
1130 IFIS="U"THEN1160      :rem 142
1140 IFST=.THEN1100      :rem 88
1160 PRINT#4,PS$:GOSUB6000:GOTO9410
      :rem 96
1170 GETBS:IFBS=""THEN1170      :rem 183
1180 GOTO1100      :rem 197
1210 IFP=0THEN1270      :rem 9
1220 IFP=2THEN1260:REM VIC      :rem 11
1230 IFC<NETHENPRINT#4,CHR$(C);:GOTO1270
      :rem 179
1240 IFC>NSTHENPRINT#4,CHR$(C-NC);:GOTO1270
      :rem 130
1250 IFC>NRTHENPRINT#4,CHR$(C+NC);:GOTO1270
      :rem 128
1260 PRINT#4,CHR$(C);      :rem 169
1270 PRINTCHR$(C);:RETURN      :rem 65
1280 :           :rem 5
2000 REM"*****"
      :rem 172
2010 REM"*{ 7 SPAZI}SP{ 2 E}D{ SPAZI}SCRI
      PT{ 8 SPAZI}*"      :rem 165
2020 REM"*****"
      :rem 174
2040 :           :rem 0
2050 PMS="SPEEDSCRIPT":FL=0:XX=48
      :rem 28
2060 PS$="" :SA=0:P=0:GOSUB6080      :rem 202
2090 OPEN1,8,15:OPEN4,4,SA:OPEN8,8,8,NA$+
      " ,P,R"      :rem 152
2100 PRINTCHS;:GOSUB6000:IPDSTHEN2060
      :rem 17
2110 PRINTCHS;:FORI=1TOXX:GET#8,AS:NEXTI
      :rem 46
2120 GET#8,AS:GOSUB2200:GETIS:IFIS="I"THE
      N2180      :rem 144
2130 IFIS="U"THEN2170      :rem 145
2140 IFST=.THEN2120      :rem 92
2170 PRINT#4,PS$,PS$:PS$="" :GOSUB6000:GOT
      O9410      :rem 215

```

LOAD

Newsletter Editor

```

2180 GETBS:IFBS=""THEN2180      :rem 187
2190 GOTO2120      :rem 202
2200 IFAS=""THEN2120      :rem 45
2210 C=ASC(AS)      :rem 210
2220 IFC=NBTHENC=NA:GOSUB28:RETURN
      :rem 225
2230 IFC>N0THENRETURN      :rem 133
2240 IFC>NFTHENC=C-NC:GOSUB28:RETURN
      :rem 90
2250 IFC<NCTHENC=C+NR:GOSUB28:RETURN
      :rem 99
2260 IFC<NETHENGOSUB28:RETURN      :rem 158
2270 C=C+NC:GOSUB28:RETURN      :rem 136
2280 :           :rem 6
3000 REM"*****"
      :rem 173
3010 REM"* LE{ 2 T}ORE{ SPAZI}FILE{ SPAZI}
      SEQUENZIALI { SPAZI}*"      :rem 169
3020 REM"*****"
      :rem 175
3030 :           :rem 0
3040 GOSUB6080:OPEN1,8,15:OPEN4,4,SA
      :rem 104
3050 OPEN8,8,8,NA$+" ,S,R":GOSUB6000:IPDST
      HEN3040      :rem 27
3060 PRINTCHS;      :rem 64
3070 GET#8,AS:IFAS=""THEN3070      :rem 62
3080 C=ASC(AS):IFFL-C=0THEN3070      :rem 9
3090 GOSUB3150:GETIS:IFIS="I"THEN3130
      :rem 146
3100 IFIS="U"THEN3120      :rem 139
3110 IFST=.THEN3070      :rem 95
3120 PRINT#4,PS$:PS$="" :GOSUB6000:GOTO941
      0      :rem 224
3130 GETBS:IFBS=""THEN3130      :rem 179
3140 GOTO3070      :rem 203
3150 IFC=94THENC=NA:GOTO3310:REM FRECCIA
      VERSO L'ALTO      :rem 250
3155 IFC=95THENC=:GOTO3310:REM FRECCIA V
      ERSO SINISTRA      :rem 105
3160 IFC<>36THEN3300:REM NON E' UN '$'
      :rem 254
3170 GET#8,AS:IFAS<>"$"THENB=C:C=36:GOSUB
      20:C=B:GOTO3300      :rem 12
3180 TT$="" :FORI=1TO2:GET#8,AS:TT$=TT$+AS
      :NEXT      :rem 228
3190 IFTT$="PR"THENC=13:GOSUB483:RETURN
      :rem 209
3200 IFTT$="NL"ORTT$="SK"THENC=13:GOTO331
      0      :rem 20
3210 IFTT$="RM"ORTT$="IN"THEN3280
      :rem 194
3220 IFTT$="TB"ORTT$="ST"THEN3280
      :rem 202
3230 IFTT$="PG"ORTT$="QC"THEN3280
      :rem 185
3240 IFLETS(TT$,1)THEN3280      :rem 29
3250 IFTT$="TI"THENC=13:GOTO3310:rem 144
3260 IFRIGHT$(TT$,1)="$"THENGET#8,AS:GOTO
      3180      :rem 85
3270 RETURN      :rem 172
3280 GOSUB3290:GOSUB3290:C=:GOTO3300
      :rem 196

```

```

3290 GET#8,A$:IFAS<>"(SPAZI)"THEN3290:RET
URN :rem 61
3300 C=ASC(A$):GOTO3310 :rem 13
3310 GOSUB28:RETURN :rem 203
3320 : :rem 2
5000 REM"*****" :rem 175
5010 REM"*{ 8 SPAZI}HES{SPAZI}WRITER
{ 9 SPAZI}*" :rem 191
5020 REM"*****" :rem 177
5030 PN$="HES/PAGE{SPAZI}MATE" :rem 126
5040 FL=34:REM IL SEPARATORE DI CAMPO DEL
FILE E' IL CARATTERE 'DOPPI APICI'
:rem 102
5050 GOTO3040 :rem 202
5060 : :rem 5
5070 REM"*****" :rem 182
5080 REM"*{ 7 SPAZI}TOTL.TEXT{ 11 SPAZI}
*" :rem 255
5090 REM"*****" :rem 184
5100 PN$="TOTL.TEXT":FL=13 :rem 44
5110 GOTO3040 :rem 199
5120 : :rem 2
5130 REM"*****" :rem 179
5140 REM"*{ 4 SPAZI}PAPERCLIP{SPAZI}CONT
ROL{SPAZI}Z{ 4 SPAZI}*" :rem 33
5150 REM"*****" :rem 181
5160 PN$="PAPERCLIP (CONTROL Z)":FL=0
:rem 52
5170 GOTO3040 :rem 205
5180 : :rem 8
5190 REM"*****" :rem 185
5200 REM"*{ 7 SPAZI}EASYSRIPT{ 2 SPAZI}
{ 3 SPAZI}{ 4 SPAZI}*" :rem 106
5210 REM"*****" :rem 178
5220 PN$="EASYSRIPT":FL=0:GOTO3040
:rem 134
5230 : :rem 4
5240 REM"*****" :rem 181
5250 REM"*{ 8 SPAZI}WORDPRO{SPAZI}3+
{SPAZI} { 3 SPAZI}{ 4 SPAZI}*" :rem 115
5260 REM"*****" :rem 183
5270 PN$="WORDPRO{SPAZI}3+":FL=0:GOTO3040
:rem 47
5280 : :rem 9
5290 REM"*****" :rem 186
5300 REM"*{ 4 SPAZI}PAPERCLIP{SPAZI}CONT
ROL{SPAZI}S{ 4 SPAZI}*" :rem 24
5310 REM"*****" :rem 179
5320 PN$="PAPERCLIP CTRL S" :rem 181
5330 FL=32:XX=48:GOTO2060 :rem 205
5340 : :rem 6
6000 REM"*****" :rem 176
6010 REM"* ESAME STATO UNITA' DISCHI
{SPAZI}*" :rem 181
6020 REM"*****" :rem 178
6030 : :rem 3
6040 DS=0:INPUT#1,N,MS$,T,S :rem 85

```

```

6045 PRINTRT$;RN$;"_NOME DEL FILE:"NA$:PRI
NTN;MS$;T;S :rem 102
6050 IFN=.THENRETURN:REM NESSUN PROBLEMA
CON L'UNITA' DISCHI :rem 29
6065 CLOSE1 :rem 120
6070 DS=1:RETURN:REM PROBLEMI CON L'UNITA
' DISCHI :rem 19
6075 : :rem 12
6080 REM"*****" :rem 184
6090 REM"*{ 6 SPAZI}SCELTA{SPAZI}DEL FIL
E{ 6 SPAZI}*" :rem 28
6100 REM"*****" :rem 177
6110 : :rem 2
6120 PRINTCH$;RN$;PN$:PRINT:PRINTM1$:PRIN
T :rem 248
6125 PRINT"IL MARGINE DESTRO E':"RM:PRINT
:rem 148
6130 P=1:PRINTRN$;M2$:INPUTP$:IFP$="N"THE
NP=0:GOTO6170 :rem 104
6140 IFP$="U"THEN9110 :rem 158
6150 PRINT:PRINTRN$;M4$:INPUTP$ :rem 2
6155 IFP$="S"THENP=2:SA=7:GOTO6170
:rem 25
6160 PRINT:PRINTRN$;M3$:INPUTSA :rem 34
6170 PRINT:PRINTRN$;"INSERIRE IL NOME DEL
FILE{SPAZI}":INPUTNA$ :rem 28
6180 PRINTCH$;:RETURN :rem 96
6190 : :rem 10
8000 : :rem 2
8070 REM"*****" :rem 151
8090 REM"*INIZIALI{ 2 Z}AZIONE{SPAZI}VAR
IABILI{SPAZI}*" :rem 81
8100 REM"*****" :rem 97
8110 { 2 SPAZI}8080 :rem 4
8120 DC=4:SA=7:BC=6:FC=7:CC=5 :rem 160
8130 NA=13:NB=31:NC=32:ND=35:NE=65:NF=95
:rem 51
8140 NG=99:NH=100:NI=102:NJ=104:NK=108
:rem 157
8150 NL=110:NM=114:NN=111:NO=116:NP=117:N
Q=122 :rem 138
8160 NR=64:NS=96:CLOSE1:CLOSEDC:CLOSE8
:rem 157
8170 RN$=CHR$(18):RT$=CHR$(13):REM RVS ON
E RET :rem 105
8180 CH$=CHR$(147):REM CLR/HOME :rem 2
8190 M1$="PREMI I PER TERMINARE"+RT$+"PRE
MI U PER USCIRE." :rem 95
8200 M2$="STAMPA IL TESTO <S/N/U>"
:rem 38
8210 M3$="INDIRIZZO SECONDARIO DELLA STAM
PANTE" :rem 40
8220 M4$="STAMPANTE CBM <S/N>" :rem 187
8230 M5$=RT$+"FINE{SPAZI}FILE -(SPAZI)PR
EMI{SPAZI}UN{SPAZI}TASTO" :rem 131
8240 : :rem 8
9000 REM"*****" :rem 145
9010 REM"*{ 5 SPAZI}MENU'{SPAZI}PRINCIPA
LE{ 6 SPAZI}*" :rem 7
9020 REM"*****" :rem 147
9030 : :rem 6
9040 IFPEEK(65532)<>34THEN9070 :REM VIC
:rem 183
9050 POKE36879,248:PRINTCHR$(158):CT=20
:rem 255
9060 GOTO9100 :rem 210

```



```

9070 IFPEEK(65532)<>226THEN9100:REM"C-64
      :rem 129
9080 POKE53289,BC:POKE53281,FC:REM"COLORI
      S-64 :rem 63
9090 SA=0:P=0:CT=64:BL=16:SL=16 :rem 43
9100 : :rem 4
9110 PRINTCHR$(14);CHR$(144);CH$;RNS$;
      :rem 89
9120 PRINT"NEWSLE{ 2 T}ER{SPAZI}EDITOR";
      :rem 76
9130 IFCT=20THENPRINT" ";:GOTO9150
      :rem 15
9140 PRINT :rem 91
9150 PRINTRNS$; :rem 91
9160 IFCT=20THENPRINT"{SPAZI}":REM 1 SPAZ
      IO :rem 90
9170 PRINT:PRINT:PRINTCC$;"DA QUALE WORD-
      PROCESSOR" :rem 187
9180 PRINT"LEGGI IL FILE?":PRINT
      :rem 198
9190 PRINT"1. QUICK BROWN FOX" :rem 117
9200 PRINT"2. WORD MANAGER" :rem 51
9210 PRINT"3. SPEEDSCRIPT" :rem 68
9220 PRINT"4. TOTL.TEXT" :rem 182
9230 PRINT"5. PAPER CLIP CTRL Z":rem 177
9240 PRINT"6. PAPER CLIP CTRL S":rem 172
9250 PRINT"7. EASYSCRIPT" :rem 13
9260 PRINT"8. WORDPRO 3+" :rem 147
9270 PRINT"9. HES / PAGE MATE":PRINT
      :rem 164
9280 PRINTRNS$;"10. FINE PROGRAMMA"
      :rem 122
9290 PRINTRNS$;"11. COLORE CURSORE"
      :rem 27
9300 IFCT=20THEN9350:REM VIC 20 :rem 243
9310 PRINTRNS$;"12. COLORE SCHERMO"
      :rem 3
9320 PRINTRNS$;"13. COLORE CORNICE"
      :rem 247
9350 PRINT:INPUT"QUALE NUMERO";A:rem 54
9355 IFA>13THEN9110 :rem 66
9360 IFA<7THENONAGOTO360,1000,2000,5070,5
      130,5290 :rem 184
9370 IFA>6THENA=A-6 :rem 104
9380 ONAGOTO5190,5240,5000,9400,9500,9470
      ,9440 :rem 146
9390 GOTO9110 :rem 217
9400 LIST20-:REM POTETE CAMBIARE IL VALOR
      E NUMERICO :rem 123
9410 CLOSE4:CLOSE8:CLOSE1:PRINTM5$;
      :rem 233
9420 GETAS:IFAS=""THEN9420 :rem 193
9430 GOTO9110 :rem 212
9440 IFCT<>64THEN9120:REM CAMBIAMENTO COL
      ORI :rem 220
9450 BC=BC+1:IFBC<BLTHENBC=0 :rem 176
9460 POKR53280,BC:GOTO9110 :rem 0
9470 IFCT<>64THEN9110 :rem 216
9480 FC=FC+1:IFFC<SLTHENFC=0 :rem 212
9490 POKE53281,FC:GOTO9110 :rem 251
9500 IPCC>7THENCC=0 :rem 122
9505 CC=CC+1 :rem 120
9510 ONCCGOSUB9530,9540,9550,9560,9570,95
      80,9590,9600 :rem 65
9520 GOTO9110 :rem 212
9530 CC$=CHR$(5):RETURN:REM BIANCO
      :rem 35
9540 CC$=CHR$(28):RETURN:REM ROSSO
      :rem 67
9550 CC$=CHR$(30):RETURN:REM VERDE
      :rem 29
9560 CC$=CHR$(31):RETURN:REM BLU:rem 140

```

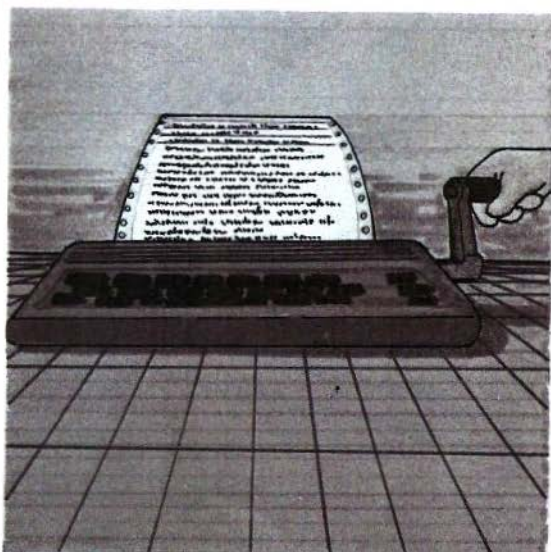


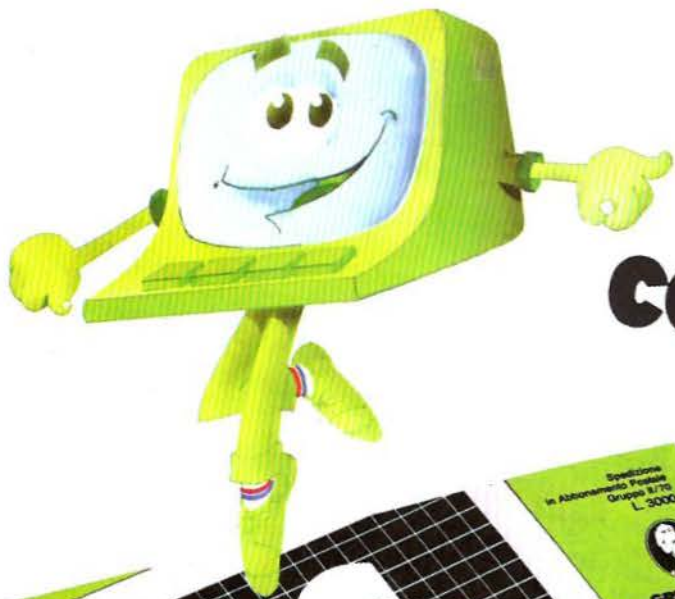
**Newsletter
Editor**

```

9570 CC$=CHR$(144):RETURN:REM NERO
      :rem 19
9580 CC$=CHR$(156):RETURN:REM PORPORA
      :rem 6
9590 CC$=CHR$(158):RETURN:REM GIALLO
      :rem 158
9600 CC$=CHR$(159):RETURN:REM BLU-VERDE
      :rem 101
9610 RETURN :rem 176
9900 REM"*****
      :rem 196
9910 REM"* IL PROGRAMMA LEGGE FILE DA *
      :rem 241
9920 REM"* WORD-PROCESSOR DIFFERENTI
      { 2 SPAZI}* :rem 55
9930 REM"* E VISUALIZZA SULLO SCHERMO *
      :rem 38
9940 REM"* O SULLA STAMPANTE IL TESTO.*
      :rem 255
9950 REM"* L'USCITA SU STAMPANTE E'
      { 3 SPAZI}* :rem 174
9960 REM"* FORMATTATA DALLA VARIABILE *
      :rem 18
9970 REM"* RM IN RIGA 1. RISPONDETE
      { 3 SPAZI}* :rem 104
9980 REM"* 'S' ALLA DOMANDA STAMPANTE *
      :rem 16
9990 REM"* CBM{SPAZI}? QUANDO POSSIBILE.
      { 4 SPAZI}* :rem 6
9999 REM"*****
      :rem 214

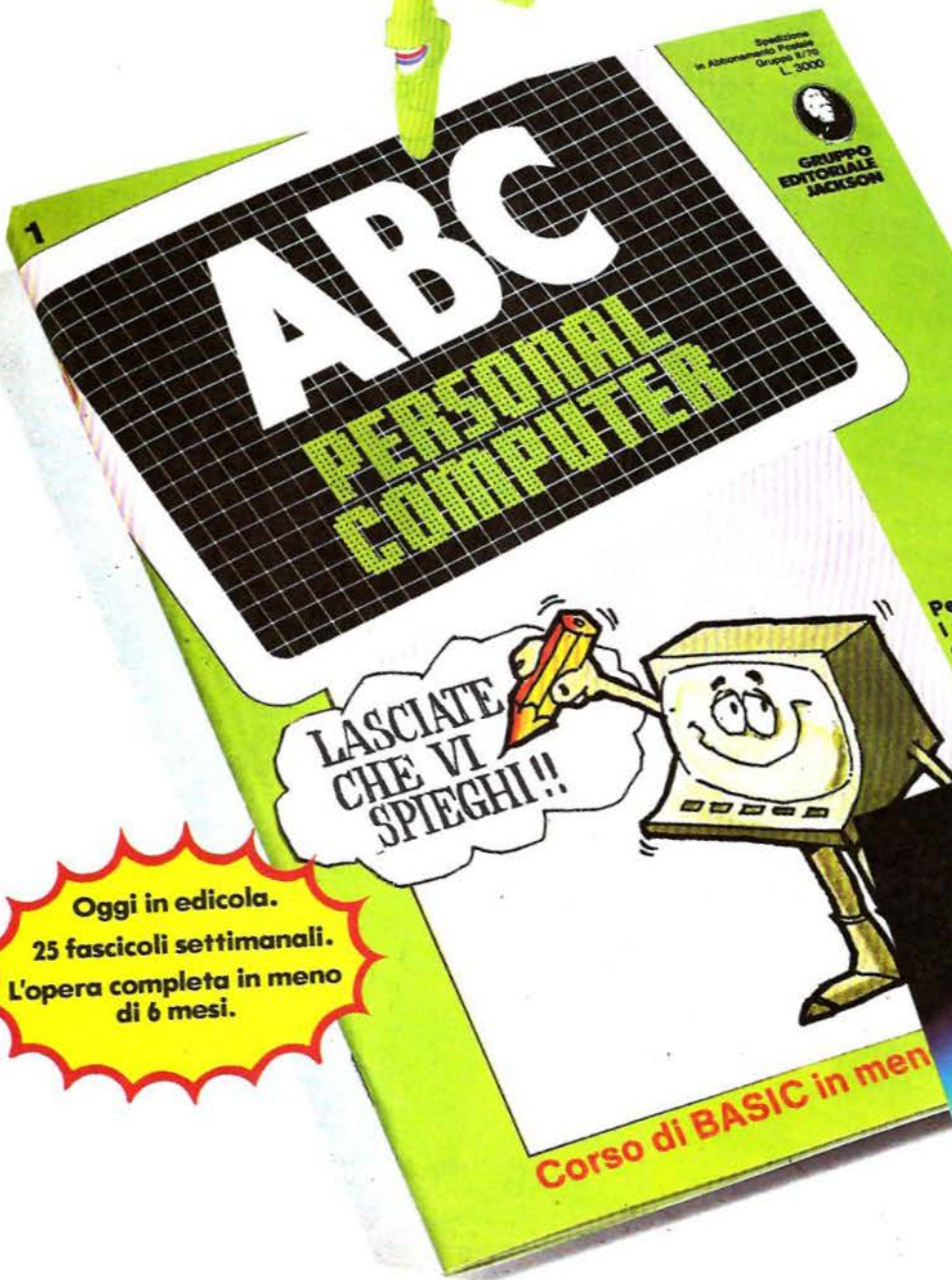
```





Diven
con

ABC
PERSONAL
COMPUTER



Spedizione
in Abbonamento Postale
Gruppo 2/70
L. 3000

GRUPPO
EDITORIALE
JACKSON

LASCIA
CHE VI
SPIEGHI!!

Perché
i personal
computer
che coe' è
un calcolatore
Micro, home,
personal

Oggi in edicola.
25 fascicoli settimanali.
L'opera completa in meno
di 6 mesi.

Corso di BASIC in men



un Jacksoniano personal computer

È in edicola ABC personal computer, il corso programmato per tutti i personal che ti insegna il Basic e a dialogare col computer in meno di 6 mesi.

ABC personal computer ti svelerà tutti i segreti dei personal. Inoltre, se non hai ancora il computer, ABC personal computer ti aiuta a sceglierlo bene, perchè ogni settimana prova per te i personal più interessanti.

ABC personal computer in soli 25 fascicoli settimanali.

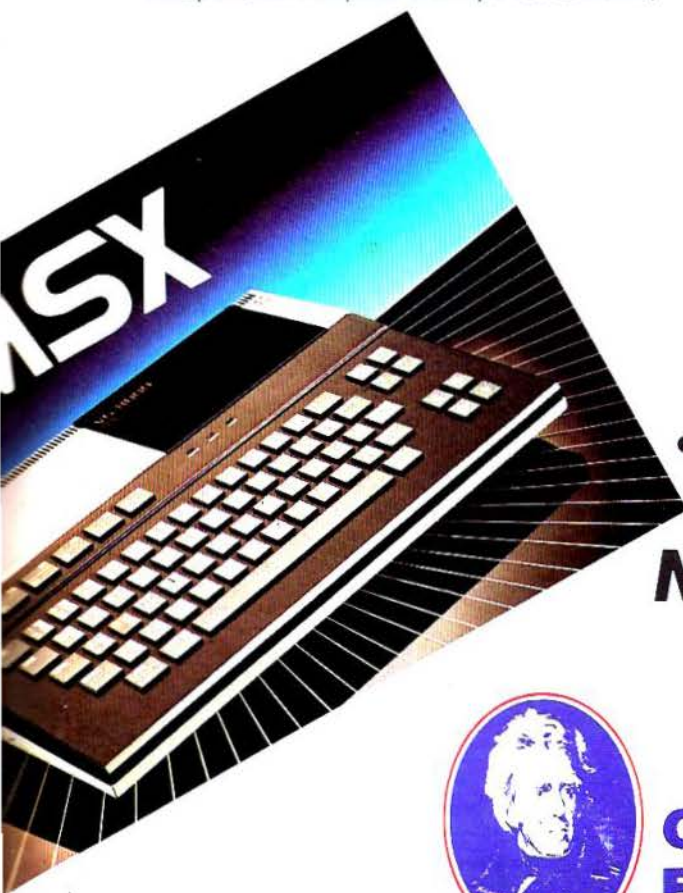
ABC personal computer è un'opera esauriente,

rapida e comprensibile a tutti, ideata e realizzata dagli specialisti del Gruppo Editoriale Jackson.

Il primo numero è in edicola, corri a prenderlo: puoi vincere 30 fantastici MSX Philips! Ma non solo.

Trovi in regalo il poster del Basic che ti consente di convertire i programmi di una macchina su qualsiasi altra.

Diventa un Jacksoniano con ABC personal computer.



**...E VINCI
30 FANTASTICI
MSX PHILIPS**



**GRUPPO
EDITORIALE JACKSON**



Labirinto 3-D

di **K. M. Queen**
trad. e adatt.
di **F. Sarcina**

In questo gioco ciò che devi fare è semplicemente uscire da un labirinto: sembra facile, ma non lo è. Partendo dall'angolo nord-ovest devi riuscire a raggiungere l'uscita, che si trova all'angolo sud-est. Inoltre, ad ogni partita il labirinto è diverso, per cui il percorso seguito una volta potrebbe, la volta successiva, portarti in un vicolo cieco. Funziona sia sul C64 che sul VIC 20 (con o senza espansione).

Dato che ti trovi in un labirinto, puoi comportarti in due modi: esplorarlo con comodo, come se stessi facendo una passeggiata, cercando di costruirne una pianta e senza porti limiti di tempo; oppure tentare di uscirne il più velocemente possibile, trasformando il gioco in una gara con te stesso e coi tuoi amici.

Dopo aver digitato e salvato il programma devi semplicemente dare il RUN: lo schermo appare vuoto, mentre il computer costruisce il labirinto, per cui devi attendere qualche secondo prima che il gioco inizi. Il tuo punto di partenza è l'angolo nord-ovest (in alto a sinistra), e il muro verso nord che hai inizialmente di fronte è contrassegnato dalla parola INIZIO. L'uscita si trova invece all'angolo sud-est (in basso a destra), in corrispondenza del muro verso sud contrassegnato dalla parola FINE. Per guardare nelle quattro direzioni possibili premi S, N, E oppure O. Durante il gioco la direzione nella quale stai puntando appare nella parte alta dello schermo. Per muoverti in tale direzione premi la barra spaziatrice. Per prendere nota delle tue mosse, ed eventualmente cercare di disegnare una mappa, ti consigliamo inoltre di fornirti di carta e matita.

Se tuttavia ad un certo punto pensi di esserti perso e di non avere molte speranze di trovare da solo l'uscita, premi A (Aiuto!): potrai in tal modo dare una breve occhiata alla mappa del labirinto, sulla quale è indicata la tua attuale posizione. Nel caso che tu non riesca ugualmente a trovare una via d'uscita potrai



comunque abbandonare premendo F e riprovare con un nuovo labirinto.

Quando sarai riuscito ad arrivare a contatto col muro contrassegnato dalle parole FINE dovrai compiere un ulteriore passo verso sud per uscire dal labirinto e concludere la partita. Ti verrà quindi comunicato il tempo impiegato per raggiungere l'uscita, e subito dopo potrai vedere la mappa del labirinto.

Funzionamento del programma

La prima cosa che il programma fa è disegnare lo sfondo, che consiste in una X gigante la quale rappresenta il corridoio visto in prospettiva, privo per ora

di incroci con altri corridoi laterali. A questo punto il computer esplora il labirinto (memorizzato in una matrice) nella direzione in cui il giocatore è rivolto e controlla l'esistenza di eventuali corridoi laterali, che vengono quindi disegnati a sinistra e a destra della X. Successivamente viene controllata la presenza di un eventuale muro di fronte all'esploratore; se esiste, tale muro viene rappresentato disegnando un rettangolo centrato sulla X, le cui dimensioni dipendono dalla distanza a cui si trova la parete. Nel caso in cui non vi sia un muro il programma prosegue la sua esplorazione del corridoio, arrivando fino al quinto spazio libero dinanzi al gio-

catore (il labirinto misura 15x15 spazi). Il risultato di questo procedimento è un disegno che simula la visione tridimensionale del corridoio del labirinto, così come apparirebbe a chi si trovasse real-

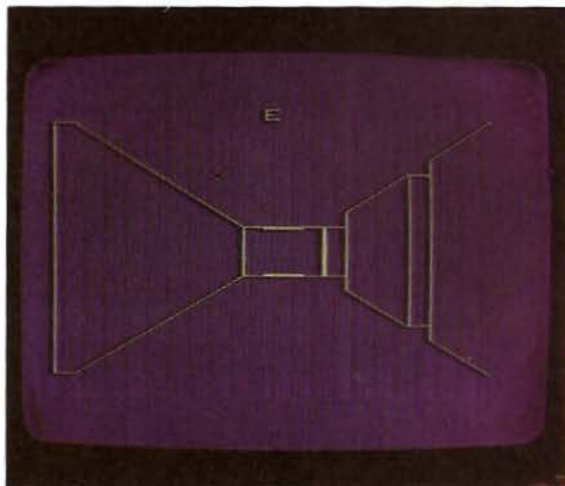
mente al suo interno. Se digiti questo programma su un VIC 20 inespanso, fai molta attenzione a non introdurre, memorizzando le linee, degli spazi extra oltre a quelli indicati

nel listado che pubblichiamo. Dato che il programma occupa pressoché tutta la memoria, anche pochi byte di troppo potrebbero causare un OUT OF MEMORY ERROR.

Labirinto 3-D

```
1 X=PEEK(648)*256:SYS65517:GP=PEEK(781):I
FGP=40THENCRS=CHRS(13) :rem 230
2 DIMM%(17,17):OS%="{GIU'}{ 2 SPAZI}{RVS}P
REMI UN TASTO":IFGP<>40THENPOKE36879,11
0 :rem 50
3 FORJ=1TO17:FORI=2TO16:M%(1,J)=1:M%(17,J
)=1:M%(I,17)=1:M%(I,1)=1:NEXT:NEXT
:rem 102
4 FORI=3TO16STEP2:FORJ=3TO16STEP2:M%(I,J)
=1:NEXT:NEXT:AJ=0:QS="N" :rem 107
5 FORK=1TO37:I=INT(RND(1)*15)+2:J=INT(RND
(1)*15)+2:M%(I,J)=1:NEXT:Q=1:B=2:C=2
:rem 244
6 FORI=1TO26:READW:READY:M%(W,Y)=0:NEXT:R
ESTORE:TI$="000000":E=0:D=-1:AI=D
:rem 62
7 POKE198,0:PRINT"{CLR}"TAB(10)QS:IF
M%(B+E,C+D)=1THEN32 :rem 35
8 PRINT"{HOME} {GIU'}M{GIU'}M{GIU'}M
{GIU'}M{GIU'}M{GIU'}M{GIU'}M{GIU'}M
{GIU'}M{GIU'}M{GIU'}M{GIU'}M{GIU'}M
{GIU'}M{GIU'}M{GIU'}M{GIU'}M{GIU'}M
{GIU'}M{GIU'}M" :rem 116
9 PRINT"{SU}N{SU}N{SU}N{SU}N{SU}N{SU}N
{SU}N{SU}N{SU}N{SU}N{SU}N{SU}N{SU}N:PR
INTTAB(13)"{SU}N{SU}N{SU}N{SU}N{SU}N
{SU}N{SU}N{SU}N{SU}N{SU}N{SU}N{SU}N
{SU}N{SU}N{SU}N{SU}N{SU}N{SU}N{SU}N
:rem 127
10 IFM%(B+AI,C+AJ)=0THENGOSUB58 :rem 88
11 IFM%(B-AI,C-AJ)=0THENGOSUB60 :rem 86
12 IFM%(B+2*E,C+2*D)=1THEN38 :rem 5
13 IFM%(B+2*E+AI,C+2*D+AJ)=0THENGOSUB52
:rem 236
14 IFM%(B+2*E-AI,C+2*D-AJ)=0THENGOSUB54
:rem 243
15 IFM%(B+3*E,C+3*D)=1THEN40 :rem 3
16 IFM%(B+4*E,C+4*D)=1THEN45 :rem 11
17 IFM%(B+4*E+AI,C+4*D+AJ)=0THENGOSUB56
:rem 248
18 IFM%(B+4*E-AI,C+4*D-AJ)=0THENGOSUB57
:rem 254
19 IFM%(B+5*E,C+5*D)=1THEN47 :rem 18
20 GOSUB48:GETQS:IFQS=""THEN20 :rem 45
21 IFQS="E"THENPRINT"{CLR}":ZS=1:GOSUB63:
GOTO76 :rem 25
22 IFQS="N"THENE=0:D=-1:AI=-1:AJ=0:GOTO7
:rem 108
23 IFQS="S"THENE=0:D=1:AI=1:AJ=0:GOTO7
:rem 24
24 IFQS="E"THENE=1:D=0:AI=0:AJ=-1:GOTO7
:rem 56
25 IFQS="O"THENE=-1:D=0:AI=0:AJ=1:GOTO7
:rem 67
26 IFQS="A"THENQS="":GOSUB62:GOTO7
:rem 31
27 IFQS="" THENQS="":GOTO29 :rem 241
28 GOTO20 :rem 5
29 IFM%(B+E,C+D)<>0THEN20 :rem 136
30 IFM%(B+2*E,C+2*D)=1THEN20 :rem 252
31 B=B+2*E:C=C+2*D:GOTO7 :rem 99
32 PRINT"{HOME}O[< 20 Y>]P"CR$:FORI=1TO1
9 :rem 13
33 PRINT"<<G>]{ 20 SPAZI}[<M>]"CR$:NEXT
:rem 240
34 PRINT"L[< 20 P>]@{HOME}" :rem 125
```

```
35 PRINT"{HOME}"TAB(10)QS:IFB+E=2ANDC+D=1
ANDQS="N"THENPRINTTAB(8)"{SU}{RVS}INIZ
IO" :rem 19
36 IFB=16ANDC=16ANDQS="S"THENPRINTTAB(9)
{WHT}{ 5 GIU'}{RVS}FINE":FORI=1TO3E3:N
EXT :rem 66
37 GOTO20 :rem 5
38 PRINT"{HOME}{ 4 GIU'}{ 4 DES}O
[< 12 Y>]P"FORI=1TO12:PRINT"[ 4 DES]
[<G>]{ 12 SPAZI}[<M>]" :rem 92
39 PRINT"{ 4 DES}L[< 12 P>]@":GOTO20
:rem 64
40 PRINT"{HOME}{ 5 GIU'}{ 5 DES}O
[< 10 Y>]P"FORI=1TO10:PRINTTAB(5)"
[<G>]{ 10 SPAZI}[<M>]" :rem 252
41 PRINTTAB(5)"L[< 10 P>]@" :rem 239
42 IFB=2ANDC=3TANDQS="N"THENPRINT"{HOME}
{ 7 GIU'}"TAB(10)"{RVS}I" :rem 83
43 IFB=16ANDC+3=17ANDQS="S"THENPRINT
{HOME}{ 7 GIU'}"TAB(10)"{RVS}F"
:rem 192
44 GOTO20 :rem 3
45 PRINT"{HOME}{ 8 GIU'}"TAB(8)"O
[< 4 Y>]P{GIU'}{ 6 SIN}[<G>]
{ 4 SPAZI}[<M>]{GIU'}{ 6 SIN}[<G>]
{ 4 SPAZI}[<M>]{GIU'}{ 6 SIN}[<G>]
{ 4 SPAZI}[<M>]" :rem 21
46 PRINTTAB(8)"[<G>]{ 4 SPAZI}[<M>]{GIU'}
{ 6 SIN}L[< 4 P>]@":GOTO20 :rem 189
47 PRINT"{HOME}"PRINTTAB(9)"{ 8 GIU'}O
[< 2 Y>]P{GIU'}{ 4 SIN}[<G>]
{ 2 SPAZI}[<M>]{GIU'}{ 4 SIN}[<G>]
{ 2 SPAZI}[<M>]{GIU'}{ 4 SIN}L
[< 2 P>]@" :rem 48
48 IFB<>16ORC<>16ORQS<>"S"THENRETURN
:rem 25
49 FORI=1TO1E3:NEXT:PRINT"{CLR}{ 3 GIU'}
"TAB(4)"CE L'HAI FATTA" :rem 126
50 PRINTTAB(4)"IN"VAL(TI$)"SECONDI":PRINT
"{ 2 GIU'}"TAB(4)"{RVS}PREMI UN TASTO"
:rem 241
51 FORI=1TO1E3:NEXT:POKE198,0:GOSUB73:PRI
NT"{CLR}":GOTO63 :rem 206
52 PRINT"{HOME}{ 4 GIU'}{ 3 DES}[<M>]
{GIU'}{ 2 SIN}[<M>]P{GIU'}{ 2 SIN}
[< 2 M>]{GIU'}{ 2 SIN}[< 2 M>]{GIU'}
{ 2 SIN}[< 2 M>]{GIU'}{ 2 SIN}
[< 2 M>]{GIU'}{ 2 SIN}[< 2 M>]{GIU'}
{ 2 SIN}[< 2 M>]{GIU'}{ 2 SIN}
[< 2 M>]{GIU'}{ 2 SIN}[< 2 M>]{GIU'}
{ 2 SIN}[< 2 M>]{GIU'}{ 2 SIN}
[< 2 M>]" :rem 132
53 PRINT"{GIU'}{ 2 SIN}[<M>]@{GIU'}
{ 2 SIN}[<M>]" :RETURN :rem 241
54 PRINT"{HOME}":PRINTTAB(17)"{ 3 GIU'}
[<G>]{GIU'}{ 2 SIN}O[<G>]{GIU'}
{ 2 SIN}[< 2 G>]{GIU'}{ 2 SIN}
[< 2 G>]{GIU'}{ 2 SIN}[< 2 G>]{GIU'}
{ 2 SIN}[< 2 G>]{GIU'}{ 2 SIN}
[< 2 G>]{GIU'}{ 2 SIN}[< 2 G>]{GIU'}
{ 2 SIN}[< 2 G>]{GIU'}{ 2 SIN}
[< 2 G>]" :rem 98
55 PRINT"{GIU'}{ 2 SIN}[< 2 G>]{GIU'}
{ 2 SIN}[< 2 G>]" :PRINTTAB(17)"L[<G>]
{GIU'}{ 2 SIN} [<G>]":RETURN :rem 123
```



```

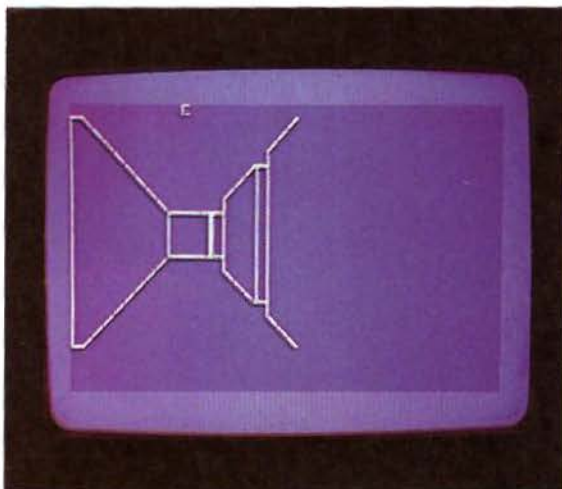
56 PRINT"{HOME}"TAB(7)"{ 8 GIU'}{<M>
   {GIU'}{ 2 SIN}{<M>}P{GIU'}{ 2 SIN}
   [< 2 M>]{GIU'}{ 2 SIN}{< 2 M>}{GIU'}
   { 2 SIN}{<M>}@{GIU'}{ 2 SIN}{<M> }":RE
   TURN :rem 205
57 PRINT"{HOME}":PRINTTAB(13)"{ 7 GIU'}
   [<G>]{GIU'}{ 2 SIN}O{<G>}{GIU'}
   { 2 SIN}{< 2 G>}{GIU'}{ 2 SIN}
   [< 2 G>]{GIU'}{ 2 SIN}L{<G>}{GIU'}
   { 2 SIN} [<G>":RETURN :rem 178
58 PRINT"{HOME} {GIU'}{SIN}P{GIU'}{SIN}
   [<M>]{GIU'}{SIN}{<M>}{GIU'}{SIN}{<M>}
   {GIU'}{SIN}{<M>}{GIU'}{SIN}{<M>}{GIU'}
   {SIN}{<M>}{GIU'}{SIN}{<M>}{GIU'}{SIN}
   [<M>]{GIU'}{SIN}{<M>}{GIU'}{SIN}{<M>}
   {GIU'}{SIN}{<M>}{GIU'}{SIN}{<M>}{GIU'}
   {SIN}{<M>}{GIU'}{SIN}{<M>}{GIU'}{SIN}
   [<M>]{GIU'}{SIN}{<M>}{GIU'}{SIN}{<M>}
   {GIU'}{SIN}{<M>}{GIU'}{SIN}@{GIU'}
   {SIN} {HOME}" :rem 242
59 RETURN :rem 78
60 FORI=X+(2*GP)TOX+(21*GP)STEPGP:POKEI,1
   01:NEXT:POKEX,32 :rem 67
61 POKEX+GP,79:POKEX+(20*GP),76:POKEX+(21
   *GP),32:RETURN :rem 81

```

```

62 PRINT"{CLR}{ 2 SPAZI}TI TROVI QUI":ZS=
   1 :rem 65
63 FORJ=1TO17:FORI=1TO17 :rem 246
64 IFM%(I,J)=1THENPRINT"{RVS} {OFF}";:GOT
   O68 :rem 169
65 IFI=BANDJ=CTHENPRINT"{CYN}Z{WHT}";:GOT
   O68 :rem 249
66 IFM%(I,J)=5THENPRINT"R";:GOTO68
   :rem 93
67 PRINT"{DES}"; :rem 150
68 NEXT:PRINT"":NEXT:FORI=1TO1E3:NEXT
   :rem 206
69 IFZSTHENZS=0:RETURN :rem 14
70 POKE198,0:PRINTOS:GOSUB73:CLR:GOTO1
   :rem 177
71 DATA2,2,2,3,2,4,4,2,4,4,4,3,16,14,16,1
   6,14,16,15,16,14,14,8,5,8,6,8,7
   :rem 209
72 DATA8,8,8,9,8,10,14,8,14,9,14,10,3,2,2
   ,5,5,2,12,10,10,6,6 :rem 23
73 GETAS:IFAS=""THEN73 :rem 247
74 IFAS="Q"THENPOKE198,0:SYS198 :rem 104
75 PRINT"{CLR}":RETURN :rem 234
76 PRINTOS:WAIT198,1:POKE198,0:SYS198
   :rem 29

```



ERRATA CORRIGE

Nel programma "Interprete Turtle Graphics per C64" pubblicato sul numero 6 di SUPERVIC, ci sono stati segnalati inconvenienti nel salvataggio su nastro delle procedure. Questi ultimi si verificano a causa di una leggera interferenza tra la routine in linguaggio macchina di gestione del raster interrupt (che permette di avere contemporaneamente visualizzati pagina grafica e testo) e la routine di salvataggio dati su nastro. Per ovviare a questo inconveniente, abbiamo modificato quest'ultimo come qui riportat. Occorre dunque modificare o aggiungere le seguenti linee al Programma 3:

```

23018 GOSUB27000:OPEN2,1,0,WD$+".TURTLE"
23060 CLOSE2:SYS49322:RETURN
24018 GOSUB27000:OPEN2,1,1,WD$+".TURTLE"
24040 CLOSE2:SYS49322:RETURN
27000 SYS65409:POKE53280,2:POKE53281,11:D
   H=0:GOSUB3000:POKE53287,0:POKESE,1
27010 PRINTCHR$(30):RETURN

```

Nessuna modifica al programma è invece necessaria se si utilizza il disk drive.

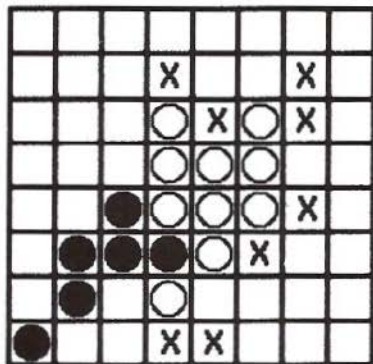


Reversi

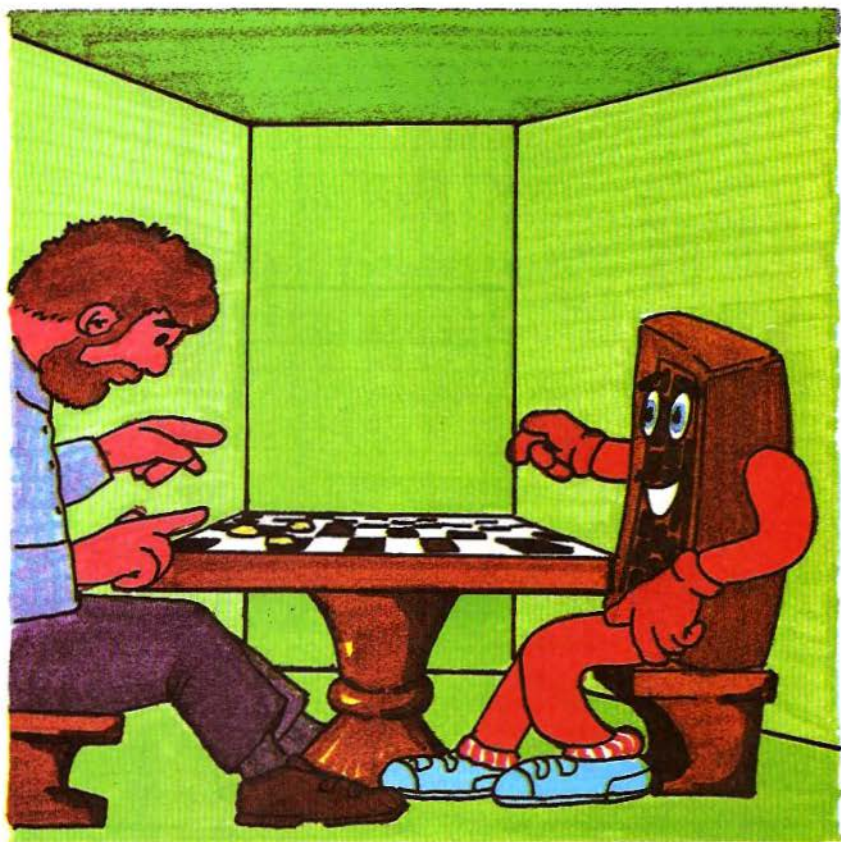
di **S. Puckett**
trad. e adatt.
di **F. Sarcina**

Tra i giochi da scacchiera, questo è uno dei più moderni e interessanti da giocare con un computer. Noi di SUPERCOMMODORE te lo proponiamo nella versione VIC 20 e C64. È richiesto un joystick (porta 2 per il C64).

Reversi ti dà la possibilità di giocare sia contro il computer che contro un altro giocatore. Le regole sono veramente semplici: si gioca sistemando una pedina alla volta sulla scacchiera, in modo da catturare le pedine avversarie. Per fare ciò bisogna chiudere una o più pedine dell'opposto colore tra due del proprio; in questo modo tutte le pedine avversarie catturate cambiano di colore e diventano in possesso del giocatore che muove. Con ogni mossa si deve catturare almeno un pezzo avversario; se ciò non è possibile, si salta il turno. Vince chi alla fine del gioco (cioè quando tutte le 64 caselle della scacchiera sono occupate o quando nessuno dei due giocatori può effettuare una mossa valida) possiede più pedine del proprio colore.



Mosse possibili per il nero



Le catture sono effettuabili in orizzontale, in verticale, nelle due diagonali o, se la mossa è particolarmente abile, anche in più direzioni contemporaneamente. Per porre la pedina sulla scacchiera premete il pulsante del joystick quando il cursore è posizionato sulla casella da voi scelta. Per quanto riguarda la miglior tattica di gioco lasciamo che il vostro computer ve la insegni pian piano, battendovi le prime volte...

Nella versione per il C64 è possibile memorizzare sulla scacchiera vuota una determinata configurazione iniziale delle pedine; la versione per il VIC ha invece la possibilità di far scontrare il computer contro se stesso, osservando la sua tattica. Il programma contiene spiegazioni particolareggiate, quindi il modo migliore per impararne l'uso è proprio provare a giocare. Buon divertimento!

Reversi versione per VIC 20

```

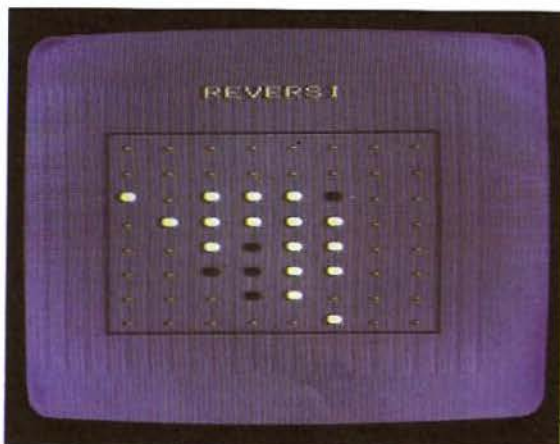
10 GOSUB540 :rem 122
20 IFJ=2ORF=64THEN470 :rem 2
30 IFC1=1THENC1=2:C2=1:GOTO50 :rem 219
40 C1=1:C2=2 :rem 100
50 IFC1=1ANDB$="C"THEN270 :rem 155
60 IFC1=2ANDW$="C"THEN270 :rem 178
70 GETA$:IFAS="P"THENJ=J+1:GOTO20 :rem 220
80 POKE37154,127:A=PEEK(37152)AND128:B=(A=0) :rem 183
90 POKE37154,255:A=PEEK(37151) :rem 147
100 R=R+((AAND8)=0)-((AAND4)=0) :rem 124
110 C=C+((AAND16)=0)-B :rem 121
120 IFR<0THENR=0 :rem 206
130 IFR>7THENR=7 :rem 223
140 IFC<0THENC=0 :rem 178
150 IFC>7THENC=7 :rem 195
160 B=8079-44*R+C+C :rem 219
170 D=PEEK(B):D1=PEEK(B+30720) :rem 156
180 POKEB+30720,C1-1:POKEB,D+128:rem 16
190 FORE=0TO99:NEXT :rem 193
200 POKEB+30720,D1:POKEB,D :rem 230
210 FORE=0TO99:NEXT :rem 186
220 IF(AAND32)=0THENP=9*(7-R)+C:GOTO240 :rem 247
230 GOTO70 :rem 53
240 IFB(P)THEN50 :rem 156
250 GOSUB400:IFNTHENA=P:GOSUB370:POKEL-30720,46:POKEL,7:B(P)=0:GOTO50:rem 1
260 J=0:F=F+1:GOTO20 :rem 131
270 M=-99:FORE=0TO70:IFB(E)THEN350 :rem 7
280 N=0:FORX=0TO7:A=E:B=0 :rem 251
290 A=A+D(X):IFA<0ORA>70THEN320 :rem 51
300 IFB(A)=C2THENB=B+1:GOTO290 :rem 2
310 IFB(A)=C1THENN=N+B :rem 29
320 NEXT:IFN=0THEN350 :rem 31
330 N=N+RND(1)*.9:IFF<55THENN=G(E)+G(E)-N :rem 96
340 IFM<NTHENM=N:P=E :rem 16
350 NEXT:IFM=-99THENJ=J+1:GOTO20:rem 250
360 J=0:F=F+1:GOSUB400:GOTO20 :rem 210
370 POKE36874,230:FORH=0TO99:NEXT:POKE36874,0 :rem 203
380 L=38491+26*INT(A/9)+A+A:POKEL,C1-1 :rem 90
390 B(A)=C1:RETURN :rem 59
400 A=P:GOSUB370:POKEL-30720,81 :rem 41
410 N=1:FORX=0TO7:A=P:B=0 :rem 2
420 A=A+D(X):IFA<0ORA>70THEN460 :rem 51
430 IFB(A)=C2THENB=B+1:GOTO420 :rem 1
440 IFB(A)<>C1ORB=0THEN460 :rem 2
450 N=0:A=P:FORE=1TOB:A=A+D(X):GOSUB370:NEXT :rem 243
460 NEXT:RETURN :rem 243
470 FORE=0TO70:IFB(E)=1THENS1=S1+1 :rem 29
480 IFB(E)=2THENS2=S2+1 :rem 68
490 NEXT:PRINT"{HOME}{WHT}":IFS1>S2THENPRINT"VITTORIA AI NERI":PRINTS1"A"S2:GOTO520 :rem 152
500 IFS1<S2THENPRINT"VITTORIA AI BIANCHI":PRINTS2"A"S1:GOTO520 :rem 178
510 PRINT"{ 6 SPAZI}PAREGGIO!" :rem 214
520 GETA$:IFAS=""THEN520 :rem 81
530 RUN :rem 141
540 FORA=0TO7:READD(A):NEXT :rem 173
550 DIMB(70),G(70):A=RND(-TI):F=4:POKE36878,15 :rem 76

```

```

560 FORA=0TO34:READB:G(A)=B:G(70-A)=B:NEXT :rem 3
570 FORA=8TO62STEP9:B(A)=3:NEXT :rem 176
580 B(30)=2:B(31)=1:B(39)=1:B(40)=2 :rem 232
590 C1=2:C2=1 :rem 158
600 POKE36879,110:C$="{BLK}NERI":GOSUB770 :rem 14
610 C$="{WHT}BIANCHI":GOSUB770:W$=A$ :rem 33
620 IFZ=0THEN690 :rem 188
630 PRINT"{CLR}{GIU'}SPOSTA IL CURSORE CON {GIU'}IL JOYSTICK." :rem 68
640 PRINT"{GIU'}PREMI 'FIRE' PER { 6 SPAZI}{GIU'}EFFETTUARE LA MOSSA," :rem 12
650 PRINT"{GIU'}'P' PER PASSARE." :rem 191
660 IFZ=2THENPRINT"{GIU'}IL COLORE DEL CURSORE {GIU'}INDICA A CHI TOCCA." :rem 60
670 PRINT"{GIU'}PREMI LO SPAZIO ..." :rem 119
680 GETA$:IFAS<>" "THEN680 :rem 156
690 PRINTCHR$(142)"{CLR}{WHT}{ 7 SPAZI}REVERSI" :rem 17
700 PRINT"{ 2 GIU'}{ 2 DES}{BLK}{<A>}{ 15 *}{<S>}" :rem 242
710 FORA=1TO8:PRINT"{BLK}{ 2 SPAZI}_ {YEL} . . . . .{BLK}_" :rem 252
720 PRINT"{ 2 DES}_"TAB(18)"_" :rem 172
730 PRINT"{SU}{ 2 DES}{<Z>}{ 15 *}{<X>}" :rem 224
740 PRINT"{HOME}{ 10 GIU'}"TAB(9)"{WHT}Q {BLK}Q" :rem 5
750 PRINTTAB(9)"{GIU'}{BLK}Q {WHT}Q :rem 244
760 RETURN :rem 125
770 PRINTCHR$(14)"{CLR}{WHT}CHI GIOCA CON I" :rem 158
780 PRINT"{GIU'}PEZZI {RVS}"C$"{OFF}{WHT}?" :rem 166
790 PRINT"{ 2 GIU'}{ 2 DES}{RVS}C{OFF}OMPUTER" :rem 96
800 PRINT"{GIU'}{ 2 DES}{RVS}U{OFF}OMO" :rem 24
810 GETA$:IFAS=""THEN810 :rem 85
820 IFAS="U"THENZ=Z+1 :rem 224
830 RETURN :rem 123
840 DATA-9,-8,1,10,9,8,-1,-10 :rem 164

```




```

850 DATA16,-4,4,2,2,4,-4,16,0,-4,-12,-2,-
2,-2,-2,-12,-4,0 :rem 189
860 DATA4,-2,4,2,2,4,-2,4,0,2,-2,2,0,0,2,
-2,2 :rem 128

```

Reversi

versione per C64

```

10 POKE56,56:CLR:TU=1:POKE53281,15:CO=542
72:SC=13:CHIP$="%&{GIU'}{ 2 SIN}#$":PL
=1 :rem 158
20 DIMBO(80),TA(71),PT(71),A(71),PO(80)
:rem 51
30 GOSUB2500 :rem 170
40 GOSUB760 :rem 129
50 GOSUB1060 :rem 172
60 GOSUB960 :rem 133
70 IFDE=1THENGOSUB1210:GOTO150 :rem 105
80 FORY=2TO5:FORX=2TO5 :rem 175
90 READA:PO(Y*9+X)=A:NEXT:NEXT :rem 159
100 BO(30)=2:BO(31)=1:BO(39)=1:BO(40)=2:B
C=2:WC=2 :rem 137
110 POKE646,1:POKE214,9:PRINT:PRINTTAB(10
):CHIP$ :rem 189
120 POKE214,9:PRINT:POKE646,0:PRINTTAB(13
):CHIP$ :rem 192
130 POKE646,0:POKE214,12:PRINT:PRINTTAB(1
0):CHIP$ :rem 232
140 POKE214,12:PRINT:POKE646,1:PRINTTAB(1
3):CHIP$ :rem 237
150 FL=1:X=4:Y=4:WC$=STR$(WC)+" ":BC$=STR
$(BC)+" " :rem 203
160 IFTU=1THENM$="{BLU}MOSSA: NERO
{ 2 SPAZI}":GOTO180 :rem 122
170 M$="{BLU}MOSSA: BIANCO" :rem 18
180 POKE214,4:PRINT:PRINTTAB(26);M$:POKE2
14,12:PRINT:PRINTTAB(31);BC$:rem 22
190 IFPL=1THENAL=BC+1:GOTO210 :rem 19
200 AL=WC+1 :rem 82
210 POKE214,18:PRINT:PRINTTAB(31);WC$
:rem 1
220 POKE214,10:PRINT:PRINTTAB(25)"PEDINE
NERE" :rem 39
230 POKE214,16:PRINT:PRINTTAB(25)"PEDINE
BIANCHE" :rem 238
240 IFCM=1ANDTU=PLTHENGOSUB1930:GOTO450
:rem 0
250 POKE53269,1 :rem 44
260 JV=PEEK(56320):FR=JVAND16:JV=15-(JVAN
D15):S=0 :rem 162
270 IFJV=1ANDY>0THENY=Y-1:GOTO320
:rem 84
280 IFJV=2ANDY<7THENY=Y+1:GOTO320
:rem 89
290 IFJV=4ANDX>0THENX=X-1:GOTO320
:rem 86
300 IFJV=8ANDX<7THENX=X+1:GOTO320
:rem 85
310 GOTO330 :rem 99
320 POKECO+4,17:POKECO+1,25:FORI=1TO20:NE
XTI:POKECO+4,16 :rem 191
330 GETA$:IFAS<>"F"THEN380 :rem 215
340 POKE214,20:PRINT:PRINTTAB(26)"VUOI VE
RAMENTE";SPC(28);"TERMINARE":rem 57
345 PRINTSPC(29)"IL GIOCO" :rem 21
350 GETA$:IFAS="S"THEN1740 :rem 218
360 IFAS<>"N"THEN350 :rem 96
370 POKE214,20:PRINT:PRINTTAB(26)"
{ 14 SPAZI}";SPC(28);"{ 9 SPAZI}"
:rem 171
375 PRINTSPC(29)"{ 8 SPAZI}" :rem 18
380 POKE53248,32*X*24:POKE53249,58+Y*24
:rem 145

```

```

390 SC=SC+1:IFSC=16THENSCL=13 :rem 202
400 POKE2040,SC :rem 75
410 IFFR=16THEN260 :rem 39
420 XY=Y*9+X:IFBO(XY)>0THEN260 :rem 84
430 POKECO+4,33:POKECO+1,10:FORJ=1TO50:NE
XTJ :rem 209
440 POKECO+4,32:FORJ=15TO0STEP-1:POKECO+1
,T:NEXT :rem 20
450 IFPL=0THEN530 :rem 238
460 POKE53269,0:POKE214,Y*3:PRINT
:rem 215
470 POKE646,TU-1:PRINTTAB(X*3+1);CHIP$
:rem 32
480 POKECO+4,33:POKECO+1,10:FORJ=1TO50:NE
XTJ :rem 214
490 POKECO+4,32:FORJ=15TO0STEP-1:POKECO+1
,T:NEXT :rem 25
500 IFPO(XY)=0THEN530 :rem 249
510 GOSUB1500 :rem 220
520 IFCHIPS>0THENGOSUB1610:BO(XY)=TU:GOTO
650 :rem 67
530 POKE214,20:PRINT:PRINTTAB(25)"{BLU}MO
SSA ILLEGALE":PRINTTAB(25)"PERDI IL T
URNO" :rem 105
540 POKECO+4,33:POKECO+1,5:FORJ=1TO300:NE
XTJ:POKECO+4,32:POKECO+1,0 :rem 115
550 FORJ=1TO150:NEXTJ :rem 53
560 IFPL=0THEN630 :rem 241
570 POKECO+4,33:POKECO+1,10:FORJ=1TO150:N
EXTJ :rem 7
580 POKECO+4,32:FORJ=15TO0STEP-1:POKECO+1
,T:NEXT :rem 25
590 POKE646,15:POKE214,Y*3:PRINT:rem 168
600 PRINTTAB(3*X+1);CHIP$ :rem 223
610 POKECO+4,33:POKECO+1,10:FORJ=1TO50:NE
XTJ :rem 209
620 POKECO+4,32:FORJ=15TO0STEP-1:POKECO+1
,T:NEXT :rem 20
630 POKE214,20:PRINT:PRINTTAB(25)"
{ 14 SPAZI}":PRINTTAB(25)"
{ 14 SPAZI}" :rem 232
640 GOTO700 :rem 106
650 IFTU=1THENBC=BC+CHIPS+1:WC=WC-CHIPS:G
OTO670 :rem 20
660 WC=WC+CHIPS+1:BC=BC-CHIPS :rem 48
670 FORQ=1TO8 :rem 30
680 IFXY+OF(Q)>-1THENPO(XY+OF(Q))=1
:rem 124
690 NEXTQ :rem 47
700 TU=3-TU :rem 134
710 IFWC=0ORBC=0ORWC+BC=64THEN1740
:rem 78
720 GOSUB750 :rem 181
730 IFXY=0ORXY=7ORXY=63ORXY=70THENGOSUB23
50 :rem 116
740 GOTO150 :rem 106
750 FORI=0TO71:TA(I)=0:NEXT:RETURN
:rem 173
760 PRINT"{CLR}{BLK}":PRINTTAB(16)"{RVS}R
EVERSI" :rem 93
770 PRINTTAB(10)"{ 2 GIU'}I (B)IANCHI INI
ZIANO" :rem 3
780 PRINTTAB(11)"I (N)ERI INIZIANO"
:rem 35
790 GETA$:IFAS="B"THEN1740:GOTO810
:rem 41
800 IFAS<>"N"THEN790 :rem 103
810 PRINTTAB(9)"{ 2 GIU'}SCHIERAMENTO (N)
ORMALE" :rem 206
820 PRINTTAB(7)"SCHIERAMENTO (R)IDISEGNAT
O" :rem 214
830 GETA$:IFAS="R"THENDE=1:GOTO850
:rem 23

```



```

840 IFA$<>"N"THEN830 :rem 102
850 PRINTTAB(13)"{ 2 GIU'}(U)N GIOCATORE" :rem 173
860 PRINTTAB(12)"(D)UE GIOCATORI" :rem 202
870 GETA$: IFA$="D"THEN950 :rem 164
880 IFA$<>"U"THEN870 :rem 117
890 CM=1:PRINTTAB(12)"{ 2 GIU'}LIVELLO? ( :rem 142
1-2)" :rem 142
900 GETA$:LE=VAL(A$):IFLE<1ORLE>2THEN900 :rem 176
910 PRINTTAB(8)"{ 2 GIU'}IL C64 GIOCA CON :rem 58
I (N)ERI" :rem 58
920 PRINTTAB(6)"IL C64 GIOCA CON I (B)IAN :rem 215
CHI" :rem 215
930 GETA$: IFA$="B"THENPL=2:GOTO950 :rem 29
940 IFA$<>"N"THEN930 :rem 104
950 RETURN :rem 126
960 A$=" [<A>]{ 2 *}[<R>]{ 2 *}[<R>] :rem 193
{ 2 *}[<R>]{ 2 *}[<R>]{ 2 *}[<R>]
{ 2 *}[<R>]{ 2 *}[<R>]{ 2 *}[<S>"
970 B$="-{ 2 SPAZI}-{ 2 SPAZI}- :rem 76
{ 2 SPAZI}-{ 2 SPAZI}-{ 2 SPAZI}-
{ 2 SPAZI}-{ 2 SPAZI}-{ 2 SPAZI}-"
980 C$=" [<Q>]{ 2 *}{+ 2 *}{+ 2 *}{+ :rem 228
{ 2 *}{+ 2 *}{+ 2 *}{+ 2 *}{+ 2 *}
 [<W>"
990 D$=" [<Z>]{ 2 C}[<E>]{ 2 C}[<E>] :rem 252
{ 2 C}[<E>]{ 2 C}[<E>]{ 2 C}[<E>]
{ 2 C}[<E>]{ 2 C}[<E>]{ 2 C}[<X>"
1000 PRINT"{CLR}{BLU}":PRINTA$ :rem 112
1010 FORI=1TO7 :rem 58
1020 PRINTB$:PRINTB$:PRINTC$ :rem 17
1030 NEXTI:PRINTB$:PRINTB$:PRINTD$; :rem 16
1040 RETURN :rem 165
1050 GOTO1050 :rem 197
1060 PRINTTAB(9)"{ 2 GIU'}MEMORIZZAZIONE :rem 136
SPRITES" :rem 136
1070 FORI=832TO1024 :rem 60
1080 READA:POKEI,A :rem 69
1090 NEXTI :rem 82
1100 POKE2040,15:POKE53287,4 :rem 27
1110 IFPEEK(14616)=63THEN1150 :rem 102
1120 POKE56334,PEEK(56334)AND254:rem 13
1130 POKE1,PEEK(1)AND251 :rem 99
1140 FORI=0TO1023:POKEI+14336,PEEK(I+5324 :rem 63
8):NEXT :rem 63
1150 POKE1,PEEK(1)OR4 :rem 207
1160 POKE56334,PEEK(56334)OR1 :rem 117
1170 FORI=14336+280TO14336+311 :rem 96
1180 READA:POKEI,A:NEXT :rem 191
1190 POKE53272,(PEEK(53272)AND240)+14 :rem 236
1200 RETURN :rem 163
1210 POKE53248,32:POKE53249,58:POKE53269, :rem 98
1
```

```

1220 PRINT"{HOME}{ 6 GIU'}{BLU}";TAB(25)" :rem 120
PREMI (N) PER":PRINTTAB(25)"PEDINE N
ERE{ 2 GIU'}" :rem 218
1230 PRINTTAB(25)"PREMI (B) PER":PRINTTAB :rem 120
(25)"PEDINE BIANCHE{ 2 GIU'}" :rem 120
1240 PRINTTAB(25)"PREMI 'SPAZIO'":PRINTTA :rem 117
B(25)"PER CASELLA" :rem 118
1245 PRINTTAB(25)"VUOTA" :rem 187
1250 FORY=0TO7:FORX=0TO7 :rem 15
1260 POKE53248,32+X*24:POKE53249,58+Y*24 :rem 191
1270 GETA$:XY=X+Y*9 :rem 118
1280 IFA$="B"THENWC=WC+1:BO(XY)=2:GOTO135 :rem 246
0 :rem 246
1290 IFA$="N"THENBC=BC+1:BO(XY)=1:GOTO135 :rem 216
0 :rem 216
1300 IFA$=" "THENPOKECO+4,17:POKECO+1,25: :rem 207
FORI=1TO20:NEXTI:POKECO+4,16:GOTO145
1310 U=U+1:IFU=6THENU=1 :rem 139
1320 IFU=1THENSC=SC+1:IFSC=16THENSC=13 :rem 117
1330 POKE2040,SC :rem 126
1340 GOTO1270 :rem 203
1350 POKE646,BO(XY)-1 :rem 181
1360 POKECO+4,33:POKECO+1,10:FORJ=1TO50:N :rem 4
EXTJ :rem 4
1370 POKECO+4,32:FORJ=15TO0STEP-1:POKECO+ :rem 71
1,T:NEXT :rem 71
1380 POKE214,Y*3:PRINT :rem 59
1390 PRINTTAB(X*3+1);CHIP$ :rem 22
1400 POKECO+4,33:POKECO+1,10:FORJ=1TO50:N :rem 255
EXTJ :rem 255
1410 FORE=1TO8 :rem 59
1420 POKECO+4,32:FORJ=15TO0STEP-1:POKECO+ :rem 67
1,T:NEXT :rem 67
1430 IFXY+OF(E)>-1THENPO(XY+OF(E))=1 :rem 142
1440 NEXTE :rem 77
1450 NEXTX:NEXTY :rem 51
1460 PRINT"{HOME}{ 6 GIU'}{BLU}";TAB(25)" :rem 254
{ 13 SPAZI}":PRINTTAB(25)"
{ 11 SPAZI}{ 2 GIU'}" :rem 254
1470 PRINTTAB(25)"{ 13 SPAZI}":PRINTTAB(2 :rem 232
5)"{ 14 SPAZI}{ 2 GIU'}" :rem 232
1480 PRINTTAB(25)"{ 14 SPAZI}":PRINTTAB(2 :rem 199
5)"{ 11 SPAZI}" :rem 199
1485 PRINTTAB(25)"{ 5 SPAZI}" :rem 50
1490 RETURN :rem 174
1500 CHIPS=0:FORI=1TO8:L=1:V=0:XX=0 :rem 165
1510 V=V+OF(I):IFXY+V>70ORXY+V<0THEN1550 :rem 227
1520 IFBO(XY+V)=5THEN1550 :rem 215
1530 IFBO(XY+V)=3-TUTHENXX=1:L=L+1:GOTO15 :rem 164
10 :rem 164
1540 IFXX=1ANDBO(XY+V)=TUTHENGOSUB1570 :rem 192
1550 NEXT :rem 10
1560 RETURN :rem 172
1570 W=1:V=0 :rem 143
1580 V=V+OF(I):TA(XY+V)=TU :rem 73
1590 W=W+1:IFW<L-1THEN1580 :rem 86
1600 CHIPS=CHIPS+W-1:RETURN :rem 236
1610 FORI=0TO71 :rem 112
1620 IF TA(I)=0ORTA(I)=5THEN1720 :rem 47
1630 POKE646,TU-1:L=INT(I/9) :rem 124
1640 POKE214,L*3:PRINT :rem 45
1650 POKECO+4,33:POKECO+1,10:FORJ=1TO15:N :rem 7
EXTJ :rem 7
```

```

1660 POKECO+4,32:FORJ=15TO0STEP-1:POKECO+
1,T:NEXTJ :rem 147
1670 PRINTTAB ( (I-9*L)*3+1 );CHIPS:rem 53
1680 POKECO+4,33:POKECO+1,10:FORJ=1TO15:N
EXTJ :rem 10
1690 POKECO+4,32:FORJ=15TO0STEP-1:POKECO+
1,T:NEXTJ :rem 150
1700 BO (I)=TU :rem 217
1710 POKECO+4,32:FORJ=15TO0STEP-1:POKECO+
1,T:NEXTJ :rem 143
1720 NEXTI :rem 82
1730 RETURN :rem 171
1740 PRINT "{HOME}":FORI=3TO25:PRINTSPC(25
)" { 15 SPAZI}";:NEXTI :rem 58
1750 IFBC<WCTHENM$="NERO":HI=BC:LO=WC:GOT
O1780 :rem 138
1760 IFBC<WCTHENM$="BIANCO":HI=WC:LO=BC:G
OTO1780 :rem 1
1770 T1=1:HI=BC:LO=WC :rem 251
1780 Z=INT (HI/6):FORY=0TOZ:FORX=26TO31
:rem 162
1790 IFX+Y*6-26=HITHENX=31:GOTO1840
:rem 103
1800 POKECO+4,33:POKECO+1,X+Y*4:FORJ=1TO5
0:NEXTJ :rem 220
1810 POKECO+4,32:FORJ=15TO0STEP-1:POKECO+
1,T:NEXT :rem 70
1820 IFX+6*Y-26<BCTHENPOKE1384+X+Y*40,81:
POKE55656+X+Y*40,0 :rem 169
1830 IFX+6*Y-26<WCTHENPOKE1384+X+7+Y*40,8
1:POKE55656+X+Y*40+7,1 :rem 132
1840 NEXT:NEXT :rem 133
1860 PRINT "{HOME} { 3 GIU' }":IFT1=1THENPRI
NTTAB (28)"PATA":GOTO1880 :rem 242
1870 PRINTTAB (27);M$;" VINCE" :rem 142
1880 PRINTTAB (27)HI;" A ";LO :rem 22
1890 PRINT "{ 5 GIU' }":PRINTTAB (26)"ANCORA
{ 2 SPAZI}S/N" :rem 23
1900 GETA$:IFA$="N"THENPOKE198,0:SYS198
:rem 67
1910 IFA$<>"S"THEN1900 :rem 201
1920 GOTO10 :rem 102
1930 HY=-32000:POKE53269,0 :rem 155
1940 HI=-32000:FORXY=0TO71 :rem 8
1950 IFBO (XY)>ORPO (XY)=0THENNEXT:GOTO204
0 :rem 181
1960 GOSUB1500:IFCHIPS=0THENNEXT:GOTO2040
:rem 106
1970 TT=WC+BC:QW=TT/8*CHIPS+PT (XY)*(65-TT
)/8 :rem 194
1980 IFLE=2ANDCHIPS=A1THENQW=10000
:rem 95
1990 IFLE=2ANDREC=0THENGOSUB2110:NEXT:GOT
O2040 :rem 161
2000 IFQW>HITHENHI=QW:H1=XY:NEXT:GOTO2040
:rem 192
2010 IFHI=0THENNEXTXY:GOTO2040 :rem 168
2020 IFQW/HI>.85ANDQW/HI<1.15THENZZ=INT (R
ND (1)*2):IFZZ=1THENHI=QW:H1=XY
:rem 31
2030 NEXT :rem 4
2040 IFLE=2ANDREC=1THENRETURN :rem 127
2050 IF (HI=-32000ANDLE=1)OR (HY=-32000ANDL
E=2)THENFL=0:CHIPS=0 :rem 122
2060 XY=H1 :rem 47
2070 IFLE=2THENXY=H2 :rem 239
2080 GOSUB750 :rem 230
2090 Y=INT (XY/9):X=XY-Y*9 :rem 31
2100 RETURN :rem 163
2110 A1=AL:FORE=0TO71 :rem 222
2120 A (E)=BO (E) :rem 0

```

LOAD

```

2130 IFTA (E)>0THENBO (E)=TA (E):A1=A1+1
:rem 99
2140 NEXTE :rem 75
2150 FORQ=1TO8 :rem 73
2160 IFXY+OF (Q)>-1THENPO (XY+OF (Q))=PO (XY+
OF (Q))+1 :rem 213
2170 NEXTQ :rem 90
2180 BO (XY)=TU :rem 68
2190 NW=QW:REC=1:Y1=XY :rem 138
2200 TU=3-TU:GOSUB1940:REC=0 :rem 188
2210 QY=NW-HI:TU=3-TU :rem 56
2220 IFQY>HYTHENHY=QY:H2=Y1 :rem 16
2230 IFHY=0THEN2250 :rem 92
2240 IFQY/HY>.85ANDQY/HY<1.15THENZZ=INT (R
ND (1)*2):IFZZ=1THENHY=QY:H2=Y1
:rem 51
2250 XY=Y1 :rem 65
2260 FORE=0TO70 :rem 109
2270 BO (E)=A (E):NEXT :rem 127
2280 GOSUB750 :rem 232
2290 FORQ=1TO8 :rem 78
2300 IFY1+OF (Q)<0THEN2330 :rem 163
2310 IFPO (Y1+OF (Q))=2THENPO (Y1+OF (Q))=1:G
OTO2330 :rem 84
2320 PO (Y1+OF (Q))=0 :rem 16
2330 NEXTQ :rem 88
2340 RETURN :rem 169
2350 IFXY=7THEN2410 :rem 116
2360 IFXY=63THEN2440 :rem 170
2370 IFXY=70THEN2470 :rem 172
2380 FORI=9TO13:PT (I)=15-I:NEXT :rem 132
2390 FORI=1TO37STEP9:PT (I)=6-INT (I/9):NEX
T :rem 108
2400 RETURN :rem 166
2410 FORI=6TO42STEP9:PT (I)=6-INT (I/9):NEX
T :rem 102
2420 FORI=16TO12STEP-1:PT (I)=I-10:NEXT
:rem 65
2430 RETURN :rem 169
2440 FORI=54TO58:PT (I)=60-I:NEXT:rem 186
2450 FORI=64TO28STEP-9:PT (I)=INT (I/9)-1:N
EXT :rem 202
2460 RETURN :rem 172
2470 FORI=61TO58STEP-1:PT (I)=I-55:NEXT
:rem 89
2480 FORI=69TO33STEP-9:PT (I)=INT (I/9)-1:N
EXT :rem 206
2490 RETURN :rem 175
2500 FORI=1TO8 :rem 64
2510 READA :rem 37
2520 OF (I)=A:NEXT :rem 239
2530 FORX=0TO71 :rem 129
2540 READA:PT (X)=A :rem 45
2550 NEXTX :rem 99
2560 FORI=8TO71STEP9:BO (I)=5:NEXT
:rem 66
2570 FORI=COTOCO+24:POKEI,0:NEXT:rem 26
2580 POKECO+5,130:POKECO+6,66:POKECO+24,1
5 :rem 194
2590 RETURN :rem 176
2600 DATA -10,-9,-8,-1,1,8,9,10 :rem 208

```

LOAD

Reversi

```

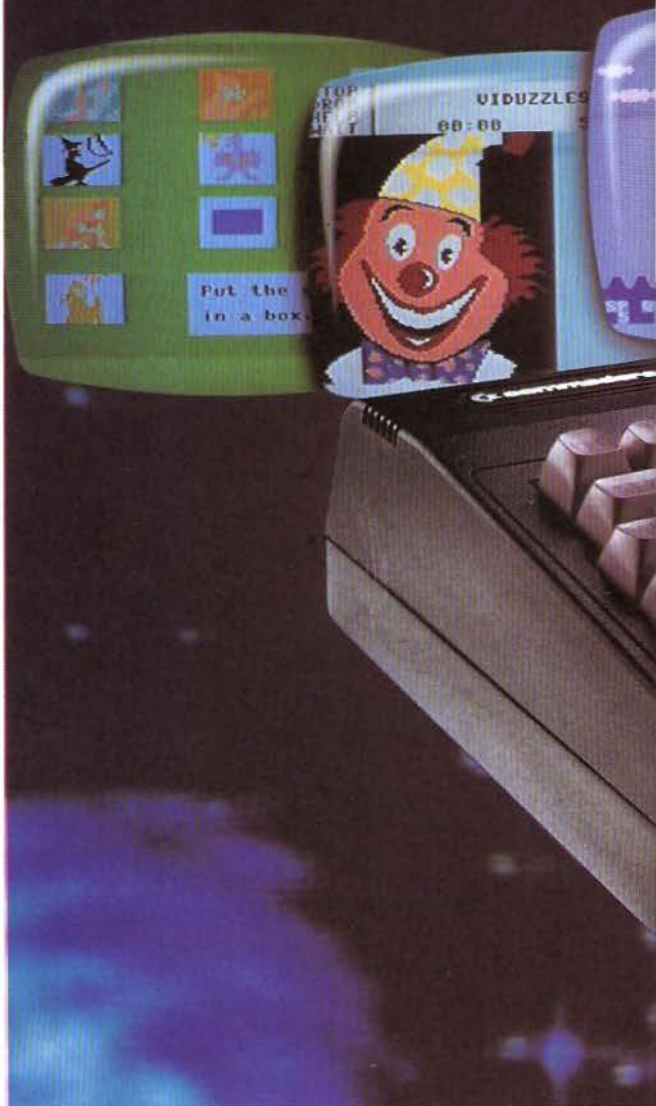
2610 DATA16,-8,5,2,2,5,-8,16,0,-8,-12,-2,
      -2,-2,-2,-12,-8,0 :rem 251
2620 DATA5,-2,8,2,2,8,-2,5,0,2,-2,2,1,1,2
      ,-2,2,0 :rem 20
2630 DATA2,-2,2,1,1,2,-2,2,0,5,-2,8,2,2,8
      ,-2,5,0 :rem 21
2640 DATA-8,-12,-2,-2,-2,-2,-12,-8,0,16,-
      8,5,2,2,5,-8,16,0 :rem 254
2650 DATA0,0,0,0,0,0,0,0 :rem 155
2660 DATA0,0,0,0,15,240,0,15 :rem 110
2670 DATA240,0,12,48,0,12,48,0 :rem 225
2680 DATA12,48,0,12,48,0,15,240 :rem 24
2690 DATA0,15,240,0,0,0,0,0 :rem 59
2700 DATA0,0,0,0,0,0,0,0 :rem 151
2710 DATA0,0,0,0,0,0,0,0 :rem 152
2720 DATA0,0,0,0,0,0,0,235 :rem 3
2730 DATA0,0,0,0,0,0,63,252 :rem 60
2740 DATA0,63,252,0,48,12,0,48 :rem 232
2750 DATA12,0,48,12,0,48,12,0 :rem 173
2760 DATA48,12,0,48,12,0,48,12 :rem 234
2770 DATA0,48,12,0,63,252,0,63 :rem 232
2780 DATA252,0,0,0,0,0,0,0 :rem 8
2790 DATA0,0,0,0,0,0,0,0 :rem 160
2800 DATA0,0,0,0,0,0,0,235 :rem 2
2810 DATA255,255,0,255,255,0,192,3
      :rem 184
2820 DATA0,192,3,0,192,3,0,192 :rem 228
2830 DATA3,0,192,3,0,192,3,0 :rem 124
2840 DATA192,3,0,192,3,0,192,3 :rem 233
2850 DATA0,192,3,0,192,3,0,192 :rem 231
2860 DATA3,0,255,255,0,255,255,0 :rem 81
2870 DATA0,0,0,0,0,0,0,0 :rem 159
2880 DATA0,0,0,0,0,0,0,8 :rem 168
2890 DATA63,63,63,63,31,15,7,0,0 :rem 82
2900 DATA252,252,252,248,240,224,0,0
      :rem 16
2910 DATA0,0,7,15,31,63,63,63 :rem 182
2920 DATA0,0,224,240,248,252,252,252
      :rem 18
2930 DATA1,1,1,1,1,0,0,1,1,0,0,1,1,1,1,1
      :rem 136

```



Commo

Per co



Arriva un home computer favoloso, l'ideale per iniziare subito e meglio.

Commodore 16: nuovo, grintoso, portentoso... e niente costoso. Commodore 16 è completo, versatile, potente (usa il BASIC 3.5, la

dore 16: nuovo, grintoso, portentoso...

minciare alla grande.

GRUPPO ETHOS



...niente costoso, L. 199.000 + IVA

versione più potente del linguaggio; è utile per sviluppare l'apprendimento del linguaggio del futuro.

È facilissimo da usare, è divertente e ti dà una mano a risolvere un sacco di problemi.

Commodore 16 è un "super-Vic" per quanto riguarda la capacità, la memoria (16K), le prestazioni.

È anche un videogioco superbo, con grafica in alta risoluzione, due generatori di suono incorporati, entu-

siasmanti cartucce giochi.

Commodore ti fa scegliere.
Commodore Italiana S.p.A.
tel. 02/618321.

commodore
COMPUTER



Caratteri giganti per C64 e VIC 20

di T. Heimarck
trad. e adatt.
di F. Sarcina

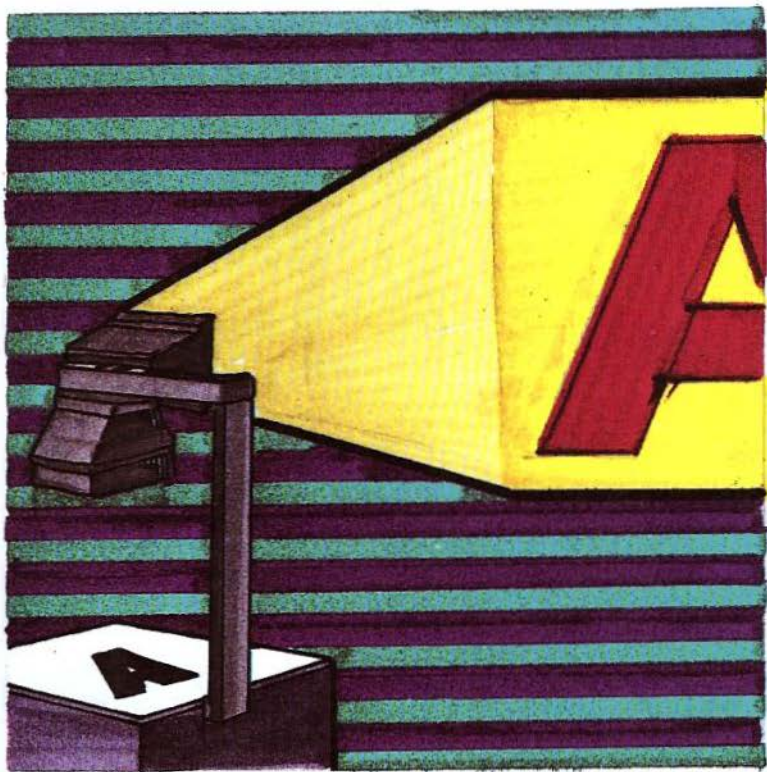
Questa breve routine in linguaggio macchina espande un carattere, facendolo apparire sullo schermo quattro volte più grande rispetto alla misura normale. I caratteri giganti possono comporre titoli o essere usati per una grande varietà di altri scopi. Il programma, che presentiamo nelle due versioni per C64 e VIC 20, è compatibile con tutte le stampanti Commodore.

Caratteri giganti possono servire per comporre titoli, per insegnare l'alfabeto ai bambini in programmi educativi o per molte altre cose ancora. Potreste costruire tali caratteri per mezzo dei simboli grafici disponibili sulla tastiera del vostro computer, ma per fare ciò sono necessarie molte prove e molto tempo; inoltre creare un intero alfabeto comporterebbe un grande spreco di memoria.

Il modo più semplice per mostrare una lettera enorme, senza fare prove laboriose o usare molta memoria, è quello di leggere, tramite istruzioni PEEK, l'area di memoria ROM generatrice di caratteri e stampare uno spazio pieno (cioè in reverse) per ogni bit che vale 1; se invece il bit vale 0, si stampa uno spazio semplice. Lo svantaggio principale di questo metodo è che ogni carattere visualizzato sarebbe otto volte più grande di uno normale, occupando un'area di ben 64 spazi, che è decisamente eccessiva: sullo schermo del VIC infatti non si riuscirebbe a stampare più di quattro lettere! Per aggirare questo problema (mantenendo comunque l'idea di leggere la ROM) possiamo tuttavia utilizzare alcuni caratteri grafici, come vedremo nel prossimo paragrafo.

I caratteri "quarto di quadrato"

Provate a battere la sequenza **IKBVDCF** tenendo premuto il tasto Commodore. I sette caratteri che appaiono sullo schermo, assieme allo spazio, formano metà del set grafico "quarto di



quadrato"; l'altra metà si ottiene premendo gli stessi tasti in modo reverse. Abbiamo così in totale 16 caratteri, uno per ogni possibile combinazione dei quattro "quarti di quadrato" accesi o spenti.

Questi simboli grafici ci permettono di costruire uno schermo a media risoluzione, meno complicato da programmare di quello ad alta risoluzione e in ogni caso più potente del sistema a bassa risoluzione illustrato precedentemente. Invece di comandare l'accensione di caratteri interi, si controllano dei grandi "pixel" (punti dello schermo), ognuno dei quali è grande un quarto di un carattere intero. In tal modo lo schermo di un VIC 20 acquista una risoluzione di

44x46 "megapunti" (ognuno dei quali è composto da $4 \times 4 = 16$ pixel), mentre la risoluzione di un C64 diventa 80x50. Questo è il concetto su cui è basato il programma che vi presentiamo. Il procedimento seguito consiste nel leggere la ROM dei caratteri e tradurre ogni bit in un "megapunto", stampando il corrispondente carattere grafico "quarto di quadrato". Si potrebbe farlo in BASIC, utilizzando una gran quantità di PEEK e di POKE, ma il linguaggio macchina è decisamente più veloce ed elegante. Il programma è facile da usare: dopo averlo digitato e salvato, battete RUN. Una breve routine in linguaggio macchina viene memorizzata tramite istruzioni POKE; per attivarla dovete effettuare

due POKE e un comando SYS:

POKE249,0 : POKE250,1 : SYS828

A questo punto dovrebbe apparire una grande "A" maiuscola, larga quattro caratteri e altrettanto alta. Provate ora a premere simultaneamente Commodore e SHIFT per ottenere il set minuscolo, tornate col cursore sulla linea contenente le POKE e premete RETURN: in tal modo apparirà una grande "a" minuscola. Se poi modificate in 129 il valore memorizzato nella locazione 250, allora la lettera gigante apparirà in reverse. Una volta che avete salvato il programma, e lo avete fatto girare, battete NEW per cancellarlo (ciò non influirà sulla routine in linguaggio macchina, che è memorizzata nel buffer di cassetta). Digitate le seguenti linee:

```
10 MK=7:REM - PER IL VIC 20
MK=3
20 PRINT"[CLR]";
30 FORX=0TO255
40 Y=(XANDMK)*4:POKE249,Y
50 IFXANDMK THENPRINT"[4 SU]";
60 POKE250,X:SYS828
70 NEXT
```

Battete RUN e l'intero set di caratteri Commodore sfilerà sullo schermo. Attenzione: non potete salvare questo programmino di esempio su nastro, in quanto ogni operazione effettuata usando il registratore cancella la routine dal buffer di cassetta.

La parte alta del carattere gigante compare nella posizione in cui si trova il cursore al momento in cui viene eseguito il comando SYS; il valore memorizzato tramite POKE nella locazione 249 determina di quanti spazi deve avanzare il cursore prima della stampa. Questo numero deve essere compreso tra 0 e 17 con il VIC oppure tra 0 e 35 con il C64.

Il numero che dovete poi memorizzare nella locazione 250 è invece il codice di schermo (non il codice ASCII) del carattere che volete stampare. I valori tra 1 e 26 sono le lettere A-Z, da 48 a 57 abbiamo le cifre 0-9, e gli altri valori inferiori a 128 corrispondono ai rimanenti simboli sulla tastiera. Da 128 a 255 ci sono gli stessi caratteri in reverse; ad esempio, se volete una S dovete usare il codice 19, mentre per ottenere la S in reverse bisogna usare il codice 19+128=147.

Dopo aver effettuato le POKE nelle locazioni 249 e 250 digitate SYS 828 e il carattere gigante apparirà istantaneamente.

Tre vantaggi e un inconveniente

Il programma in linguaggio macchina fa apparire sullo schermo gli appropriati caratteri "quarto di quadrato", usando la routine PRINT (indirizzo \$FFD2) del sistema operativo Kernal e non memorizzando direttamente i caratteri nella mappa di schermo. Con questo metodo è necessario commutare più volte tra REVERSE ON e REVERSE OFF durante la stampa, e ciò rende l'operazione piuttosto laboriosa. Tuttavia ci sono dei grossi vantaggi usando la routine in \$FFD2: vediamo.

Il primo consiste nel fatto che la versione per il VIC 20 non deve tener conto di quali espansioni di memoria siano eventualmente connesse. Se invece utilizzissimo la mappa di schermo, andremmo incontro a grossi problemi, poiché essa viene rilocata usando alcune espansioni.

Un altro vantaggio è che i caratteri giganti possono essere inviati ad una qualsiasi stampante Commodore, effettuando solo un piccolo cambiamento nel programma che memorizza la routine (vedi la linea 951 nel programma del C64 e la linea 923 in quello del VIC); questo cambiamento consiste nel sostituire un valore in un'istruzione DATA, in modo tale che per posizionare la lettera gigante vengano stampati degli spazi anziché dei cursori a destra. Per azionare la stampante digitate

```
OPEN4,4:CMD4:POKE249,xx:POKE250,yy:SYS828
```

Ricordatevi di sostituire a xx la posizione in cui volete che avvenga la stampa e a yy il codice di schermo. Se potete, regolate a zero la spaziatura tra le righe della stampante, in modo che non ci sia spazio tra i caratteri grafici che compongono la lettera gigante. Quando avete finito di stampare usate le istruzioni PRINT #4:CLOSE4 per chiudere correttamente il canale con la stampante. Sfortunatamente le stampanti non consentono movimenti di cursore verso l'alto, per cui è possibile stampare un solo carattere gigante per riga. Per aggirare questa limitazione potreste muovere manualmente la carta all'indietro, oppure usare un programma che stampi su carta una copia del contenuto dello schermo.

Il terzo vantaggio è dato dalla flessibilità della routine PRINT del Kernal: potete stampare lettere giganti in qualsiasi punto dello schermo e con qualsiasi colore (semplicemente modificando il colore del cursore). È infine possibile anche

mescolare caratteri maiuscoli, minuscoli e simboli grafici nella medesima schermata.

L'unico piccolo inconveniente è che ogni riga termina con un carattere CR (a capo), e a causa di ciò non è possibile stampare una lettera gigante all'estremità destra dello schermo.

Come funziona

Il programma BASIC comprende due cicli di istruzioni POKE. Il primo memorizza nelle locazioni 688-703 i codici ASCII modificati dei caratteri grafici "quarto di quadrato". I valori sono modificati perché contengono un flag (il bit 6) che, se vale 1, segnala che il carattere deve comparire in reverse; infatti il codice ASCII (a differenza di quello di schermo) non distingue tra i caratteri normali e quelli in reverse. Prima di stampare il carattere il bit 6 viene posto a zero effettuando un AND del codice con \$BF (191).

Il secondo ciclo memorizza la routine in linguaggio macchina nell'area che inizia dalla locazione 828 (quindi nel buffer di cassetta). La routine è scritta in modo da poter essere rilocata: se avete bisogno del buffer per un altro programma in linguaggio macchina, oppure se avete la necessità di usare il registratore, allora potete memorizzare la routine in qualsiasi altra zona di RAM (il primo ciclo non può però essere spostato). Se usate la memoria BASIC, fate attenzione a proteggere la routine contro una eventuale cancellazione da parte dei programmi BASIC.

Per chi si interessa di linguaggio macchina diamo una breve spiegazione del funzionamento della routine. Innanzitutto viene stabilito quale set di caratteri si usa (maiuscolo/grafico oppure minuscolo/maiuscolo), modificando di conseguenza un puntatore in pagina zero. Il valore del codice di schermo viene quindi moltiplicato per 8 e addizionato al puntatore, ottenendo l'indirizzo relativo al carattere desiderato. A questo punto i byte della ROM-caratteri vengono letti due per volta; effettuando alternativamente degli shift a sinistra dei byte (ASL) e delle rotazioni a sinistra dell'accumulatore (ROL), viene generato un numero compreso tra 0 e 15, che serve ad indicare (puntando all'elemento appropriato della tavola 688-703) quale dei 16 caratteri "quarto di quadrato" debba essere utilizzato. Successivamente si testa il bit 6 (reverse sì/no) ed infine viene effettuato un JSR \$FFD2 per stampare il carattere. La routine ricomincia quindi il ciclo per ricavare i successivi bit da decodificare.

Caratteri giganti versione per VIC 20

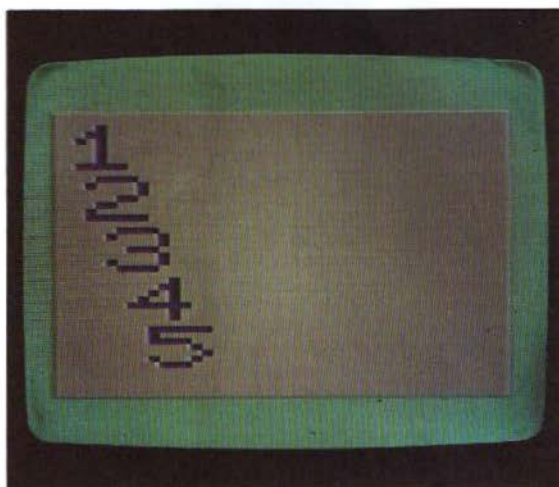
```
10 T=0:FORJ=688TO703:READK:T=T+K:POKEJ,K:
NEXT :rem 134
15 IFT<>3078THENPRINT"ERRORE NELLE
{ 10 SPAZI}ISTRUZIONI DATA":END
:rem 31
```

```
20 T=0:FORJ=828TO978:READK:T=T+K:POKEJ,K:
NEXT :rem 145
25 IFT<>17289THENPRINT"ERRORE NELLE
{ 10 SPAZI}ISTRUZIONI DATA":END
:rem 89
30 POKE249,0 :rem 141
```

```

688 DATA32,188,190,226,172,225,191,251
      :rem 148
696 DATA187,255,161,236,162,254,252,96
      :rem 158
828 DATA169,128,133,004,173,005 :rem 46
834 DATA144,041,002,240,004,169 :rem 31
840 DATA136,133,004,169,000,162 :rem 32
846 DATA003,006,250,042,202,208 :rem 28
852 DATA250,024,101,004,133,004 :rem 19
858 DATA165,250,133,003,169,000 :rem 40
864 DATA133,250,169,005,133,002 :rem 36
870 DATA160,000,177,003,133,005 :rem 26
876 DATA230,003,177,003,133,006 :rem 34
882 DATA230,003,198,002,240,028 :rem 36
888 DATA162,004,169,000,006,006 :rem 39
894 DATA042,006,006,042,006,005 :rem 30
900 DATA042,006,005,042,164,250 :rem 24
906 DATA153,048,002,230,250,202 :rem 28
912 DATA208,232,240,210,160,000 :rem 19
918 DATA166,249,240,008,169 :rem 118
923 DATA029:REM 032 USANDO LA STAMPANTE
      :rem 10
924 DATA032,210,255,202,208,250 :rem 30
930 DATA169,004,133,006,185,048 :rem 45
936 DATA002,170,189,176,002,133 :rem 43
942 DATA005,041,064,240,005,169 :rem 36
948 DATA018,032,210,255,165,005 :rem 41
954 DATA041,191,032,210,255,169 :rem 44
960 DATA146,032,210,255,200,198 :rem 40
966 DATA006,208,221,169,013,032 :rem 41
972 DATA210,255,192,016,208,196 :rem 52
978 DATA096 :rem 97
1000 PRINT"{CLR}{ 2 GIU' }{BLU}X = NUMERO
      (0-17)" :rem 231
1001 PRINT"{GIU'}Y = NUMERO (0-255)"
      :rem 90
1003 PRINT"{ 4 GIU' }POKE 249,X (POSIZIONE
      )":PRINT"{GIU'}POKE 250,Y (CARATTERE
      )" :rem 143
1004 PRINT"{GIU'}SYS828 PER ATTIVARE"

```

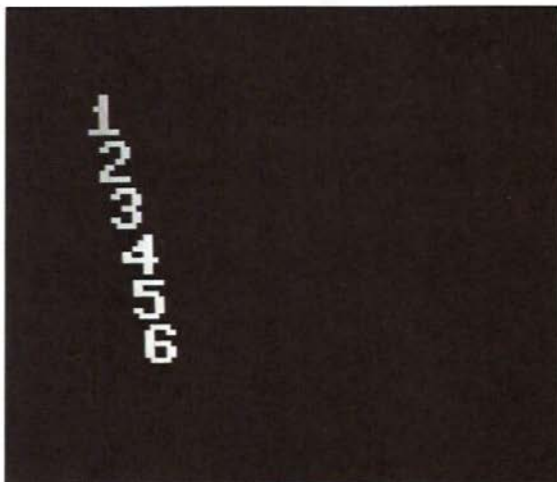


Caratteri giganti versione per C64

```

10 T=0:FORJ=688TO703:READK:T=T+K:POKEJ,K:
  NEXT :rem 134
15 IFT<>3078THENPRINT"ERRORE NELLE ISTRU
  ZIONI DATA":END :rem 31
20 T=0:FORJ=828TO1006:READK:T=T+K:POKEJ,K
  :NEXT :rem 176

```



```

25 IFT<>20306THENPRINT"ERRORE NELLE ISTRU
  ZIONI DATA":END :rem 73
30 POKE249,0 :rem 141
688 DATA32,188,190,226,172,225,191,251
      :rem 148
696 DATA187,255,161,236,162,254,252,96
      :rem 158
828 DATA169,208,133,004,173,024 :rem 46
834 DATA208,041,002,240,004,169 :rem 32
840 DATA216,133,004,169,000,162 :rem 31
846 DATA003,006,250,042,202,208 :rem 28
852 DATA250,024,101,004,133,004 :rem 19
858 DATA165,250,133,003,173,014 :rem 40
864 DATA220,041,254,141,014,220 :rem 27
870 DATA165,001,041,251,133,001 :rem 23
876 DATA169,000,133,250,169,005 :rem 46
882 DATA133,002,160,000,177,003 :rem 26
888 DATA133,005,230,003,177,003 :rem 36
894 DATA133,006,230,003,198,002 :rem 36
900 DATA240,028,162,004,169,000 :rem 28
906 DATA006,006,042,006,006,042 :rem 25
912 DATA006,005,042,006,005,042 :rem 20
918 DATA164,250,153,048,002,230 :rem 38
924 DATA250,202,208,232,240,210 :rem 26
930 DATA165,001,009,004,133,001 :rem 20
936 DATA173,014,220,009,001,141 :rem 28
942 DATA014,220,160,000,166,249 :rem 33
948 DATA240,008,169 :rem 229
951 DATA029:REM 032 USANDO LA STAMPANTE
      :rem 11
952 DATA032,210 :rem 14
954 DATA255,202,208,250,169,004 :rem 45
960 DATA133,006,185,048,002,170 :rem 38
966 DATA189,176,002,133,005,041 :rem 46
972 DATA064,240,005,169,018,032 :rem 43
978 DATA210,255,165,005,041,191 :rem 46
984 DATA032,210,255,169,146,032 :rem 47
990 DATA210,255,200,198,006,208 :rem 43
996 DATA221,169,013,032,210,255 :rem 43
1002 DATA192,016,208,196,096 :rem 153
1003 PRINT"{CLR}{ 5 GIU' }{WHT}"TAB(9)"X =
      NUMERO TRA 0 E 35" :rem 86
1004 PRINTTAB(9)"{GIU'}Y = NUMERO TRA 0 E
      255" :rem 108
1005 PRINTTAB(9)"{ 3 GIU' }POKE249,X
      { 2 SPAZI } (POSIZIONE)" :rem 142
1006 PRINTTAB(9)"{GIU'}POKE250,Y
      { 2 SPAZI } (CARATTERE)" :rem 65
1007 PRINTTAB(9)"{GIU'}SYS828{ 2 SPAZI }PE
      R ATTIVARE{ 2 GIU' }" :rem 21

```


Deflector per VIC 20

di **F.J. Tyniw**

trad. ed adatt.

di **S. Anticoli e M. Albarelli**

In questo gioco bisogna colpire, mediante una pallina vagante, dei bersagli in numero selezionabile.

Premendo il tasto recante la freccia rivolta a sinistra verrà visualizzato uno "specchietto" inclinato. La stessa funzione è assolta dal tasto 'F1' che, però, visualizza uno "specchietto" inclinato nella direzione inversa rispetto a quella visualizzata premendo la freccetta.

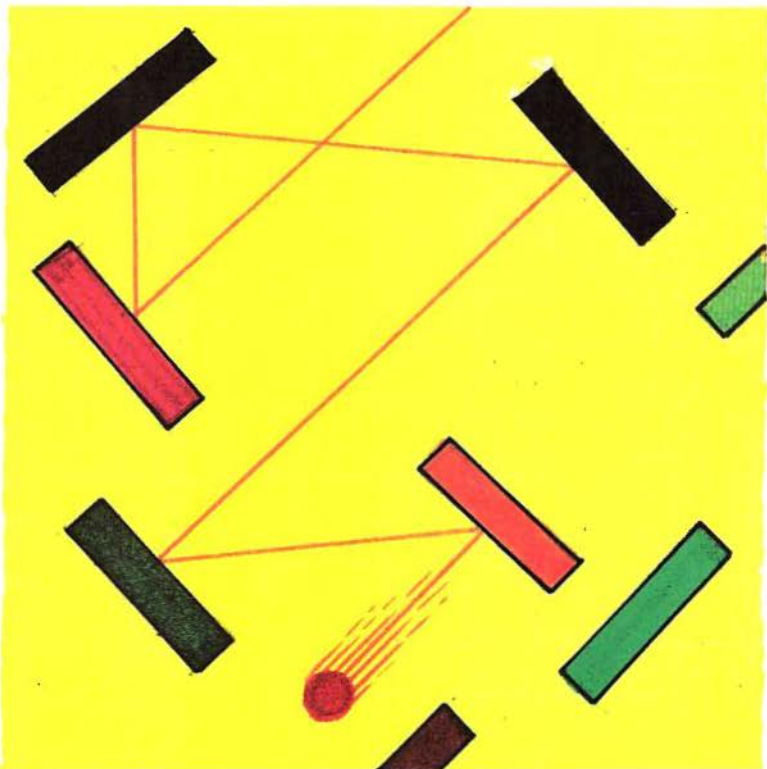
Quando la pallina tocca uno di questi "specchietti" viene riflessa di 90 gradi in una direzione che varia a seconda dell'orientamento dello "specchietto". Se la pallina colpisce un bersaglio, vengono guadagnati 10 punti.

Ogni volta che la pallina viene riflessa da uno specchietto viene sottratto un punto dal punteggio finora totalizzato. In questo gioco è anche possibile cancellare tutti gli specchietti presenti sullo schermo, premendo il tasto recante il simbolo della Lira.

Attenzione, però, perché per ogni specchietto cancellato vengono sottratti 5 punti dal punteggio totale e perciò questa opzione è meglio utilizzarla il meno possibile.

All'inizio del programma è possibile spostare lo schermo nelle quattro direzioni cardinali, in modo da centrarlo perfettamente sul proprio televisore.

E allora.....buona rimbalzata!



Deflector per VIC 20

```

10 PRINT "{CLR}":TR=208:J=3:BC=36879:VO=BC
  -1:S4=BC-2:S3=BC-3:S2=BC-4:S1=BC-5
      :rem 4
20 GOSUB63000:POKEBC,93:V=15      :rem 148
122 PRINT "{GIU'} ISTRUZIONI {DES}? (S/N)"
      :rem 20
123 GETV$:IFV$="" THEN 123         :rem 121
125 IFV$="S" THEN GOSUB1000        :rem 229
130 PRINT "{CLR}"CHR$(142)        :rem 225
140 K=0:T=0:CL=5                  :rem 131
142 INPUT "{GIU'} QUANTI BERSAGLI";J:J=ABS(
  J)                                :rem 78
144 IFJ>506 THEN PRINT "TROPPI!!!" :GOTO142
      :rem 150
146 IFJ<10 OR J>200 THEN PRINT "{GIU'} BRAVO! G
  IUSTA IDEA"                       :rem 88

```

```

155 FORI=1 TO 1000:NEXT:PRINT "{CLR}":GOSUB7
  000                                :rem 53
157 SS=7680:SR=38400                :rem 112
160 FORI=1 TO J                      :rem 34
170 A=INT(506*RND(1))                :rem 125
180 IFPEEK(SS+A)=TR THEN 170         :rem 90
185 POKES,0:POKES,0                 :rem 148
190 POKES+A,TR:POKESR+A,6:GOSUB4300
      :rem 10
200 NEXTI                            :rem 26
205 POKES2,0:POKES,0                :rem 191
210 A=INT(506*RND(1))                :rem 120
230 U=A+SS                            :rem 57
240 DI=1:IFRND(1)>.5 THEN DI=-1      :rem 184
300 GETX$                             :rem 239
310 IFX$<>" " THEN 600                :rem 34
320 NE=U+DI                           :rem 114
330 IFABS(DI)=1 THEN 430              :rem 13
340 IFDI>0 THEN 380                   :rem 235

```

LOAD**Deflector
per VIC 20**

```

350 IFNE<SSTHENDI=-DI:GOSUB6000:GOTO320
      :rem 215
355 A=NE
      :rem 174
360 IFPEEK(A)=77THENDI=-1:NE=NE-1
      :rem 220
370 IFPEEK(A)=78THENDI=1:NE=NE+1:GOTO530
      :rem 186
375 GOTO530
      :rem 112
380 IFNE>SS+506THENDI=-DI:GOSUB6000:GOTO3
20
      :rem 162
390 A=NE
      :rem 173
400 IFPEEK(A)=77THENDI=1:NE=NE+1
      :rem 168
410 IFPEEK(A)=78THENDI=-1:NE=NE-1
      :rem 217
420 GOTO530
      :rem 103
430 IFDI>0THEN490
      :rem 237
440 IFNE-22*INT(NE/22)=1THENDI=-DI:GOSUB6
200:GOTO320
      :rem 136
450 A=NE
      :rem 170
460 IFPEEK(A)=77THENDI=-22:NE=NE+DI
      :rem 106
470 IFPEEK(A)=78THENDI=22:NE=NE+DI
      :rem 63
480 GOTO530
      :rem 109
490 IFNE-22*INT(NE/22)=2THENDI=-DI:GOSUB6
200:GOTO320
      :rem 136
500 A=NE
      :rem 166
510 IFPEEK(A)=77THENDI=22:NE=NE+DI
      :rem 57
520 IFPEEK(A)=78THENDI=-22:NE=NE+DI
      :rem 104
530 POKEU,32
      :rem 173
540 IFPEEK(NE)=32THENPOKENE,81:U=NE:GOTO3
00
      :rem 190
550 IFPEEK(NE)=TRTHENK=K+1:SC=SC+10
      :rem 162
552 IFPEEK(NE)=TRTHENGOSUB5000
      :rem 139
555 POKENE,170:U=NE:FORI=1TO25:NEXT
      :rem 223
560 IFK>J-1THEN700
      :rem 33
570 GOTO300
      :rem 104
600 IFX$=""THENA=78:GOTO630
      :rem 169
610 IFX$="{F1}"THENA=77:GOTO630
      :rem 207
615 IFX$=""THENGOSUB2000
      :rem 245
616 IFX$="Q"THEN990
      :rem 75
620 GOTO320
      :rem 102
625 GOSUB4600
      :rem 231
630 IFPEEK(U+DI)=32THENPOKEU+DI,A:SL=SL+1
:SC=SC-1
      :rem 55
640 GOTO300
      :rem 102
700 REM
      :rem 123
712 PRINT"{CLR}":POKEBC,125
      :rem 176
715 IFSC>HSTHENHS=SC:PRINT"{RVS}NUOVO";
      :rem 39
716 PRINT"RECORD:"HS"{SIN} "
      :rem 228
717 PRINT"{ 6 SPAZI}LA PALLINA"
      :rem 254
720 PRINT"{GIU'}HA PRESO"SL"'SPECCHI'"
      :rem 189
730 PRINT"{GIU'}HA PRESO"J"BERSAGLI"
      :rem 101
905 PRINT"{GIU'}PUNTEGGIO:";SC
      :rem 61

```

```

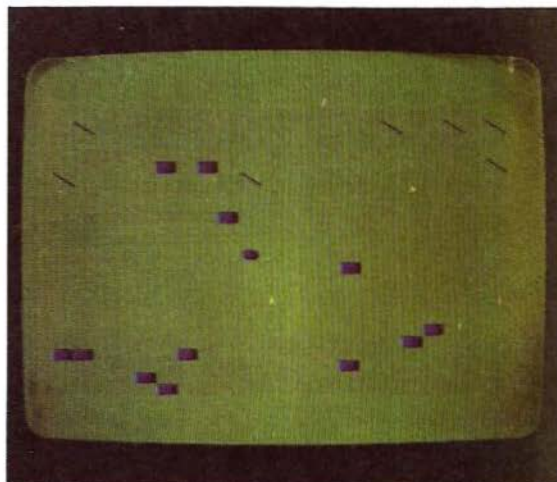
910 PRINT"{ 2 GIU'}ANCORA(S O N)"
      :rem 130
920 GETW$:IFW$=""THEN920
      :rem 133
925 IFW$="N"THEN990
      :rem 74
926 SL=0:SC=0
      :rem 234
930 PRINT:PRINT"QUANTI BERSAGLI";:INPUTJ
      :rem 158
940 J=ABS(INT(J))
      :rem 209
960 PRINT"{CLR}":POKEBC,93:GOSUB7000:K=0:
T=0:GOTO155
      :rem 5
990 PRINT"{CLR}":POKEBC,27:END
      :rem 154
1000 PRINT"{CLR}"
      :rem 37
1010 PRINTCHR$(14);"{ 3 SPAZI}L'OBBIETTIV
O{ 2 SPAZI}DI"
      :rem 252
1015 PRINT"{ 2 SPAZI}QUESTO GIOCO E' DI"
      :rem 227
1020 PRINT"{ 2 SPAZI}MANDARE LA PALLINA"
      :rem 26
1025 PRINT"{ 2 SPAZI}CONTRO I{ 2 SPAZI}BE
RSAGLI"
      :rem 0
1030 PRINT"USANDO 'F1'{ 2 SPAZI}E ' ' PER
";
      :rem 184
1035 PRINT" CREARE GLI 'SPECCHI'"
      :rem 117
1040 PRINT" E ' ' PER CANCELLARE"
      :rem 54
1045 PRINT" GLI 'SPECCHI'{ 2 SPAZI}DALLO"
      :rem 48
1050 PRINT"{ 7 SPAZI}SCHERMO."
      :rem 214
1083 PRINT"{ 2 GIU'}PREMERE UN TASTO ....
"
      :rem 181
1090 GETB$:IFB$=""THEN1090
      :rem 185
1100 PRINT"{CLR}{ 2 SPAZI}PER OGNI BERSAG
LIO"
      :rem 82
1110 PRINT"COLPITO SI CONQUISTANO";
      :rem 217
1120 PRINT"10 P.TI , 1 P.TO VIENE";
      :rem 65
1130 PRINT"{ 2 SPAZI}SOTTRATTO PER OGNI"
      :rem 126
1150 PRINT" RIFLESSIONE E 5 P.TI"
      :rem 240
1160 PRINT" VENGONO SOTTRATTI PER
{ 4 SPAZI}OGNI{ 2 SPAZI}'SPECCHIO'
{ 4 SPAZI}ANNULLATO DURANTE LA"
      :rem 45
1165 PRINT"{ 4 SPAZI}CANCELLAZIONE."
      :rem 132
1170 PRINT"{ 2 GIU'}UN TASTO PER GIOCARE.
."
      :rem 39
1180 GETA$:IFA$=""THEN1180
      :rem 183
1190 RETURN
      :rem 171
2000 FORI=SSTOSS+506
      :rem 228
2010 IFPEEK(I)<>77ANDPEEK(I)<>78THEN2030
      :rem 104
2020 GOSUB4300:POKES,0:POKES3,0:SC=SC-5:P
OKEI,32
      :rem 182
2030 NEXTI
      :rem 77
2040 RETURN
      :rem 166
4300 SO=INT(RND(1)*100)+129
      :rem 202
4310 POKEVO,V:POKES3,SO:POKES2,SO:FORTI=1
TO35:NEXTT1:RETURN
      :rem 206
5000 POKEVO,V:FORS=128TO250STEP10
      :rem 62
5010 POKES4,S
      :rem 251
5020 NEXTS
      :rem 89
5030 POKEVO,V:POKES3,250:FORII=1TO25:NEXT
II:POKES3,0:POKEVO,0:RETURN
      :rem 213
6000 POKEVO,V:POKES3,250:FORII=1TO25:NEXT
II:POKES3,0:POKEVO,0:RETURN
      :rem 211
6200 POKEVO,V:POKES3,245:FORII=1TO25:NEXT
II:POKES3,0:POKEVO,0
      :rem 191

```

```

6210 RETURN :rem 169
7000 FORI=38400TO38905:POKEI,6:NEXT:RETUR
N :rem 134
63000 REM :rem 221
63010 POKE36879,24:PRINT"{CLR}":H=PEEK(36
864):V=PEEK(36865) :rem 249
63020 PRINT"LO SCHERMO E' OK S/N "
:rem 78
63030 GETA$:IFAS=""THEN63030 :rem 27
63040 IFAS="N"GOTO63060 :rem 248
63050 PRINT"{CLR}{BLK}";:RETURN :rem 71
63060 PRINT"{ 2 GIU'}USARE I TASTI CRSR P
ER :rem 63
63070 PRINT"{ 2 SPAZI}MUOVERE LO SCHERMO
:rem 126
63080 PRINT"{GIU'}{ 2 SPAZI}E <D> QUANDO
E' OK { 2 GIU'}" :rem 214
63081 PRINT"{RVS}{RED}ROSSO{ 13 SPAZI}"
:rem 151
63082 PRINT"{RVS}{CYN}CIANO{ 13 SPAZI}"
:rem 239
63083 PRINT"{RVS}{PUR}PORPORA{ 11 SPAZI}
" :rem 166
63084 PRINT"{RVS}{GRN}VERDE{ 13 SPAZI}"
:rem 124
63085 PRINT"{RVS}{BLU}BLU{ 15 SPAZI}"
:rem 235
63086 PRINT"{RVS}{YEL}GIALLO{ 12 SPAZI}"
:rem 64
63090 GETA$:IFAS=""THEN63090 :rem 39
63100 IFAS="D"THENPRINT"{CLR}{BLK}";:RETU
RN :rem 43

```



```

63110 IFAS="{SU}"THENV=V-1:IFV<0THENV=0
:rem 244
63120 IFAS="{GIU'}"THENV=V+1:IFV>40THENV=
40 :rem 221
63130 IFAS="{SIN}"THENH=H-1:IFH<0THENH=0
:rem 202
63140 IFAS="{DES}"THENH=H+1:IFH>17THENH=1
7 :rem 187
63150 POKE36864,H:POKE36865,V:GOTO63090
:rem 25

```

ENTE FIERE SCANDIANO (RE)

4^a MOSTRA DELL' ELETTRONICA E TELECOMUNICAZIONI

SCANDIANO (RE)
28 APRILE - 1 MAGGIO 1985

TELEFONO 0522/857436/850278



Traffic per C64

di M. Celio Zanollo

Traffic è un gioco di "simulazione urbana" la cui idea è nata constatando quali pericoli costituiscono quotidianamente le automobili nei confronti degli inermi pedoni.

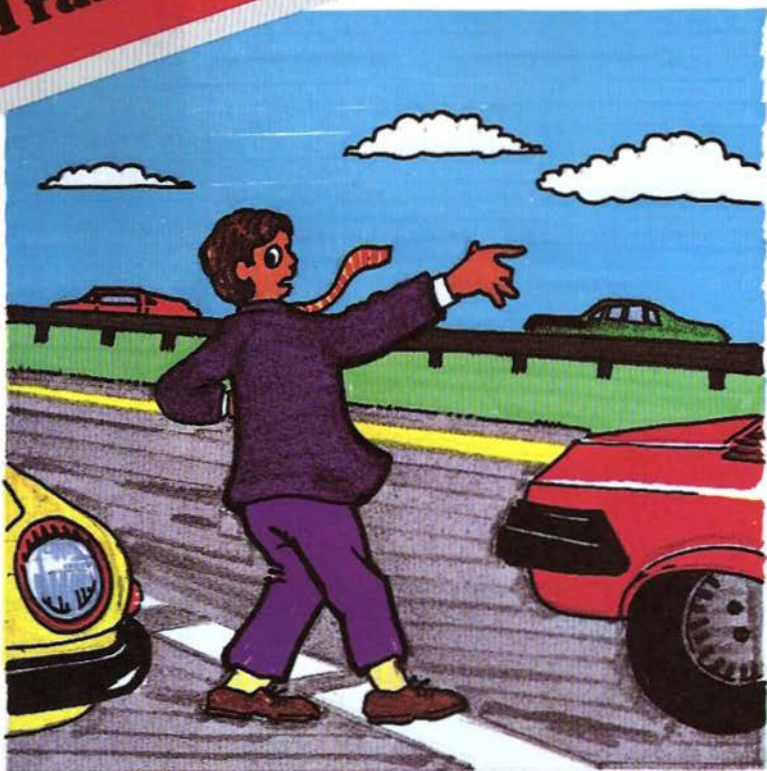
Avrete infatti notato che gli automobilisti non si preoccupano affatto dei pedoni, anche se camminano sulle strisce a loro riservate, e spesso li obbligano a fare i salti mortali per evitare investimenti.

Nel giuoco, quindi, il protagonista è un pedone che cerca disperatamente di attraversare la strada per poter giungere alla propria abitazione.

Egli ha però la sfortuna di abitare in una città di "incivili", per cui non sperate che gli automobilisti cerchino di scansarlo; dovrete pensarci voi e vi assicuro che non è per niente facile.

Se riuscirete a giungere dall'altra parte della strada, guadagnerete, accompagnati da una dolce musicchetta, 10 punti per attraversamento; in caso contrario al quinto urto con le auto perderete una vita e il suono di un'ambulanza sottolineerà il triste evento...

Se giungerete ad un punteggio equivalente a 100 punti, guadagnerete una vita e passerete al secondo schema, nel



quale l'operazione diverrà veramente complessa; noi stessi l'abbiamo superata a fatica, comunque vi facciamo i nostri migliori auguri.

Per giocare è richiesto il joystick, in PORT 2, tramite il quale potrete comandare direttamente le richieste di inizio e fine giuoco.

Traffic per C64

```

10 POKE53280,0:POKE53281,0           :rem 182
15 PRINT"{CLR}{ 2 GIU' }{DES}{RVS}{RED}M
   { 36 SPAZI}N{OFF}"                 :rem 100
20 FORI=1TO20:PRINT"{DES}{RVS} {OFF}"SPC(
   36){RVS} {OFF}":NEXT              :rem 210
25 PRINT"{DES}{RVS}N{ 36 SPAZI}M{OFF}"
                                       :rem 148
30 PRINT"{HOME}{ 3 GIU' }{ 2 DES}{RVS}
   {YEL}M{ 34 SPAZI}N{OFF}"           :rem 145
40 FORI=1TO18:PRINT"{ 2 DES}{RVS} {OFF}"S
   PC(34){RVS} {OFF}":NEXT           :rem 246
45 PRINT"{ 2 DES}{RVS}N{ 34 SPAZI}M{OFF}"
                                       :rem 179
50 PRINT"{HOME}{ 4 GIU' }{ 3 DES}{RVS}
   {BLU}M{ 32 SPAZI}N{OFF}"           :rem 66
60 FORI=1TO16:PRINT"{ 3 DES}{RVS} {OFF}"S
   PC(32){RVS} {OFF}":NEXT           :rem 17
  
```

```

65 PRINT"{ 3 DES}{RVS}N{ 32 SPAZI}M{OFF}"
                                       :rem 210
150 PRINT"{HOME}{ 7 GIU' }{WHT}{ 5 DES}
   { 2 C}{<R>}{ 2 C} [<A>]{ 2 C}I U
   { 2 C}I U{ 3 C} C[<R>]C U{ 2 C}"
                                       :rem 25
160 PRINT"{ 7 DES}B{ 3 SPAZI}{<Q>}{<R>}
   K [<Q>]{ 2 C}[<W>] [<Q>]{ 2 C}
   { 3 SPAZI}B{ 2 SPAZI}B"          :rem 121
170 PRINT"{ 4 DES}{ 3 SPAZI}J{ 3 SPAZI}J
   J{ 2 SPAZI}J{ 2 SPAZI}J{ 4 SPAZI}
   C[<E>]C J{ 3 C}":PRINT"{ 2 GIU' }#TAB(
   78)"B Y"                            :rem 128
175 PRINT"{ 2 GIU' }"TAB(8)"M A R C O
   { 2 SPAZI}Z A N O L L O"          :rem 197
180 PRINT"{ 2 GIU' }"TAB(9)"PREMI FUOCO PE
   R GIOCARE"                          :rem 11
185 IPPEEK(56320)=111THENGOTO190:rem 117
186 GOTO185                              :rem 118
190 GOSUB1000                            :rem 219
  
```

```

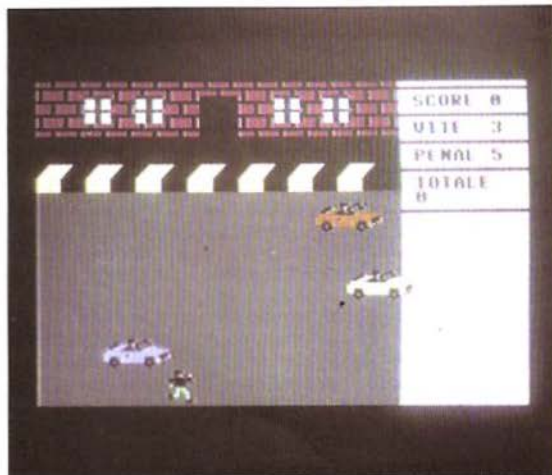
200 V=53248:POKEV+37,10:POKEV+38,0:POKEV+
28,255 :rem 217
205 FORX=704TO766:READQ:POKEX,Q:NEXT
:rem 82
210 POKE2040,11:POKEV+39,5 :rem 205
215 FORX=832TO894:READQ:POKEX,Q:NEXT
:rem 87
220 POKE2041,13:POKEV+40,5 :rem 201
225 FORX=896TO958:READQ:POKEX,Q:NEXT
:rem 99
230 POKE2042,14:POKEV+41,7:POKE2044,14:PO
KEV+42,7:POKE2046,14:POKEV+45,8
:rem 195
235 FORX=960TO1022:READQ:POKEX,Q:NEXT
:rem 123
240 POKE2043,15:POKEV+43,14:POKE2045,15:P
OKEV+44,14:POKE2047,15:POKEV+46,8
:rem 43
245 A=100:B=220:E=160:F=160:G=136:H=160:I
=190:L=200:M=166:N=200 :rem 179
250 O=150:P=120:Q=126:R=120:VI=3:rem 25
255 C=56320:D=56321:I=I+VE:M=M+VE:O=O+60:
Q=Q+60:E=E+20:G=G+20 :rem 238
260 VE=INT(20*RAND(1))+40 :rem 45
319 IFPEEK(C)=126 THENB=B-20:GOSUB490:IFB
<80THENB=70 :rem 85
320 IFPEEK(C)=119 THENA=A+40:GOSUB490:IFA
>210THENA=220 :rem 165
325 IFPEEK(C)=125 THENB=B+20:GOSUB490:IFB
>170THENB=220 :rem 174
330 IFPEEK(C)=123 THENA=A-40:GOSUB490:IFA
<20THENA=30 :rem 63
360 IFPEEK(V+30)AND10=10THENA=100:B=220:P
E=PE+1:GOSUB8000:IFPE>5THENGOSUB9000
:rem 167
370 IFE>240THENE=40:G=16 :rem 124
375 IFI>240THENI=40:M=16 :rem 143
380 IFO>240THENO=40:Q=16 :rem 155
385 IFA=140ANDB=70THENS=SC+10:GOSUB10000
:A=100:B=220 :rem 96
450 POKEV+0,A:POKEV+1,B:POKEV+2,A:POKEV+3
,B :rem 131
455 POKEV+4,E:POKEV+5,F:POKEV+6,G:POKEV+7
,H:POKEV+8,I:POKEV+9,L :rem 222
460 POKEV+10,M:POKEV+11,N:POKEV+12,O:POKE
V+13,P:POKEV+14,Q:POKEV+15,R:rem 22
465 POKEV+21,253 :rem 120
470 PRINT" {HOME} {GIU' } {RVS}"TAB(35)SC:PRI
NT" {GIU' } {RVS}"TAB(35)VI:PRINT" {GIU' }
{RVS}"TAB(35)PE :rem 120
475 PRINT" { 2 GIU' } {RVS}"TAB(29)PU:GOTO25
5 :rem 236
490 POKEV+21,254:POKE2041,13:RETURN
:rem 81
500 DATA0,15,0,0,15,0,0,13,4,0,1,4,0,15,1
2,0,255,252,0,255,252,0,207,192
:rem 180
505 DATA0,79,192,0,79,192,0,10,128,0,10,1
28,0,8,160,0,8,160,0,232,32 :rem 30
510 DATA0,232,32,0,192,32,0,192,32,0,0,60
,0,0,60,0,0,0 :rem 70
515 RESTORE :rem 191
520 DATA0,15,0,0,15,0,0,13,0,0,1,0,0,15,0
,0,63,192,0,255,192,0,207,244
:rem 76
525 DATA0,207,244,0,79,192,0,74,128,0,10,
128,0,8,128,0,8,128,0,10,128:rem 89
530 DATA0,10,128,0,8,0,0,8,0,0,8,0,0
,15,0,0,15,0,0,0,0 :rem 50
535 DATA192,192,0,195,192,0,76,48,0,112,4
8,0,48,42,0,12,170,160,254,170,170
:rem 146

```

```

540 DATA170,170,168,170,170,172,170,170,6
2,170,2,61,170,62,180,170,254,156
:rem 105
545 DATA170,206,112,0,207,0,0,252,0,0,48,
0,0,0,0,0,0,0,0,0,0,0 :rem 110
550 REM*****DATI AUTO PARTE II*****
:rem 181
555 DATA0,0,3,0,0,3,0,3,3,0,15,168,0,62,1
63,0,58,143,0,170,15,0,170,170
:rem 147
560 DATA2,170,170,2,170,174,2,128,170,2,1
40,170,2,191,170,0,51,170 :rem 192
565 DATA0,51,192,0,63,0,0,12,0,0,0,0,0,0,0
,0,0,0,0,0,0 :rem 40
570 DATA0,0,128,0,0,128,0,0,128,0,3,192,0
,3,192 :rem 13
1000 REM*****SFONDO***** :rem 102
1005 PRINT" {CLR} {RVS} {RED} {<R>} {<E>} {<R>}
{<E>} {<R>} {<E>} {<R>} {<E>} {<R>} {<E>}
{<R>} {<E>} {<R>} {<E>} {<R>} {<E>} {<R>}
{<E>} {<R>} {<E>} {<R>} {<E>} {<R>} {<E>}
{<R>} {<E>} {<R>} {<E>} {<R>} {OFF}"
:rem 6
1010 PRINT" {RVS} {RED} {<E>} {<R>} {<E>} {<R>}
{RVS} {WHT} PO {OFF} {RVS} {RED} {<E>}
{<R>} {OFF} {RVS} {WHT} PO {OFF} {RVS}
{RED} {<E>} {<R>} {<E>} {OFF} { 3 SPAZI }
{RVS} {RED} {<E>} {<R>} {<E>} {OFF} {RVS}
{WHT} PO {OFF} {RVS} {RED} {<R>} {<E>}
{OFF} {RVS} {WHT} PO {OFF} {RVS} {RED}
{<R>} {<E>} {<R>} {<E>} {OFF}" :rem 27
1020 PRINT" {RVS} {RED} {<R>} {<E>} {<R>} {<E>}
{RVS} {WHT} PO {OFF} {RVS} {RED} {<R>}
{<E>} {OFF} {RVS} {WHT} PO {OFF} {RVS}
{RED} {<R>} {<E>} {<R>} {OFF} { 3 SPAZI }
{RVS} {RED} {<R>} {<E>} {<R>} {OFF} {RVS}
{WHT} PO {OFF} {RVS} {RED} {<E>} {<R>}
{RVS} {WHT} PO {OFF} {RVS} {RED} {<E>}
{<R>} {<E>} {<R>} {OFF}" :rem 140
1025 PRINT" {RVS} {RED} {<E>} {<R>} {<E>} {<R>}
{<E>} {<R>} {<E>} {<R>} {<E>} {<R>} {<E>}
{<R>} {<E>} {OFF} { 3 SPAZI } {RVS} {RED}
{<E>} {<R>} {<E>} {<R>} {<E>} {<R>} {<E>}
{<R>} {<E>} {<R>} {<E>} {<R>} {<E>} {OFF}"
:rem 31
1030 PRINT" { 2 GIU' } {RVS} {YEL} {<E>} {OFF} {<E>}
{RVS} {<E>} {OFF} {<E>} {RVS} {<E>} {OFF} {<E>}
{RVS} {<E>} {OFF} {<E>} {RVS} {<E>} {OFF}
{<E>} {RVS} {<E>} {OFF} {<E>}
:rem 15

```



```

1035 PRINT"{RVS}{YEL}{ 2 SPAZI}{OFF}
{ 2 SPAZI}{RVS}{ 2 SPAZI}{OFF}
{ 2 SPAZI}{RVS}{ 2 SPAZI}{OFF}
{ 2 SPAZI}{RVS}{ 2 SPAZI}{OFF}
{ 2 SPAZI}{RVS}{ 2 SPAZI}{OFF}
{ 2 SPAZI}{RVS}{ 2 SPAZI}{OFF}
{ 2 SPAZI}{RVS}{ 2 SPAZI}{OFF}"
:rem 180
1040 FORI=1TO16:PRINT"{RVS}{<4>
{ 29 SPAZI}{OFF}":NEXT
:rem 44
1045 PRINT"{HOME}"TAB(29)"{RVS}{WHT}
{ 10 C}{OFF}"
:rem 204
1050 FORI=1TO23:PRINTTAB(29)"{RVS}{WHT}
{ 10 SPAZI}{OFF}":NEXT
:rem 44
1055 PRINT"{HOME}{GIU}"TAB(30)"{RVS}{SCOR
E":PRINTTAB(29)"{RVS}{ 10 C}"
:rem 107
1060 PRINTTAB(30)"{RVS}VITE"
:rem 109
1065 PRINTTAB(29)"{RVS}{ 10 C}"
:rem 224
1070 PRINTTAB(30)"{RVS}PENAL":PRINTTAB(29
)"{RVS}{ 10 C}"
:rem 244
1075 PRINTTAB(30)"{RVS}TOTALE":PRINT"
{GIU}"{RVS}"TAB(29)"{RVS}{ 10 C}"
:rem 185
2000 RETURN
:rem 162
3000 REM****RICHIESTA NUOVO GIOCO****
:rem 197
3005 A=255:B=170
:rem 51
3010 PRINT"{HOME}{ 12 GIU}"TAB(33)"{RVS}
SI":PRINT"{ 6 GIU}"TAB(33)"{RVS}NO"
:rem 230
3015 PRINT"{ 2 GIU}"TAB(31)"{RVS}ANCORA?
":PRINT"{HOME}{ 8 GIU}"{RVS}"TAB(29)
PU+SC
:rem 66
3020 IFPEEK(C)=126THENB=B-5:IFB=145THENPO
KEV+21,0:RUN
:rem 228
3025 IFPEEK(C)=125THENB=B+5:IFB=190THENPO
KEV+21,0:PRINT"{CLR}":END
:rem 102
3030 POKEV+0,A:POKEV+1,B:POKEV+2,A:POKEV+
3,B:GOTO3020
:rem 232
7000 REM****VINCITA****
:rem 135
7005 POKE54296,15:POKE54295,0
:rem 101
7010 VV=54272:POKEVV+6,9:POKEVV+5,9
:rem 106
7015 POKEVV+1,20:POKEVV+4,17
:rem 154
7020 FORNN=1TO500:NEXTNN
:rem 7
7025 POKEVV+4,0
:rem 100
7030 E=160:F=150:G=136:H=150:I=190:L=190
:M=166:N=190
:rem 148
7035 PRINT"{HOME}{ 19 GIU}"
:rem 246
7040 PRINT"{RVS}{YEL}{ 2 SPAZI}{OFF}{RVS}
{OFF}{RVS}{ 2 SPAZI}{OFF}{RVS}{ 2 SPAZI}{OFF}
{RVS}{ 2 SPAZI}{OFF}{RVS}{ 2 SPAZI}{OFF}
{RVS}{ 2 SPAZI}{OFF}{RVS}{ 2 SPAZI}{OFF}
{RVS}{ 2 SPAZI}{OFF}{RVS}{ 2 SPAZI}{OFF}
{RVS}{ 2 SPAZI}{OFF}"
:rem 175
7045 PRINT"{RVS}{YEL}{ 2 SPAZI}{OFF}{RVS}
{OFF}{RVS}{ 2 SPAZI}{OFF}{RVS}{ 2 SPAZI}{OFF}
{RVS}{ 2 SPAZI}{OFF}{RVS}{ 2 SPAZI}{OFF}
{RVS}{ 2 SPAZI}{OFF}{RVS}{ 2 SPAZI}{OFF}
{RVS}{ 2 SPAZI}{OFF}"
:rem 157
7050 PRINT"{RVS}{ 29 SPAZI}{OFF}"
:rem 65
7055 PRINT"{RVS}{ 29 SPAZI}{OFF}{WHT}":RE
TURN
:rem 101
8000 REM****SCONTRO****
:rem 162
8005 POKE54296,15:POKE54295,0
:rem 102
8010 VV=54272:POKEVV+6,0:POKEVV+5,12
:rem 140
8015 POKEVV+1,20:POKEVV+4,33
:rem 153
8020 FORNN=1TO500:NEXTNN
:rem 216
8025 POKEVV+4,0
:rem 101
8030 IFVI=0THENGOTO3000
:rem 145
8035 RETURN
:rem 176
8053 RETURN
:rem 176

```

```

9000 REM *** SIRENA ****
:rem 183
9010 VV=54272:POKEVV+6,0:POKEVV+5,237
:rem 198
9020 POKEVV+2,200:POKEVV+3,0
:rem 144
9025 AA=55:BB=42
:rem 90
9030 POKEVV+1,60:POKEVV+4,65
:rem 160
9040 FOR NN=1TO20
:rem 195
9050 POKEVV+1,AA
:rem 179
9060 FORMM=1TO50:NEXTMM
:rem 217
9070 POKEVV+1,BB
:rem 183
9080 FORMM=1TO50:NEXTMM
:rem 219
9085 IF NN=6THEN AA=53:BB=40
:rem 41
9090 NEXT NN
:rem 173
9100 POKEVV+4,0:VI=VI-1:PE=0:RETURN
:rem 201
10000 REM *** HOME MUSIC****
:rem 119
10005 POKE54296,15:POKE54295,0
:rem 143
10010 VV=54272:POKE VV+6,0:POKE VV+5,8
:rem 138
10020 PP=1000:QQ=2+(1/12)
:rem 78
10030 FOR KK=0TO4
:rem 181
10040 FOR JJ=0TO11
:rem 226
10050 IFJJ=1ORJJ=3ORJJ=6ORJJ=8ORJJ=10THEN
10080
:rem 121
10060 POKE VV+1,PP/256:POKEVV,PP AND 255
:rem 22
10070 POKEVV+4,0:POKEVV+4,65:FORNN=1TO10:
NEXTNN
:rem 221
10080 PP=PP*QQ
:rem 66
10090 IFKK=4THEN 10110
:rem 178
10100 NEXT JJ,KK
:rem 135
10110 FORNN=1TO200:NEXTNN
:rem 46
10120 POKE VV+4,0
:rem 138
10130 IFSC=100THENGOSUB7000:PU=PU+100:SC=
0:VI=VI+1
:rem 41
10140 RETURN
:rem 214

```

é in edicola

VIDEO Giochi

é in edicola

Una pubblicazione
GRUPPO EDITORIALE JACKSON



Roulette per C16

di M. Cristuib Grizzi

Ecco la versione per C16 del programma "Roulette per C64" pubblicata sul numero 2 di SUPERVIC dello scorso anno. Data la rapida diffusione del C16 - almeno a giudicare dalle richieste di programmi pervenutici da parte dei lettori - ci proponiamo, compatibilmente con la reperibilità degli stessi programmi, di pubblicare giochi ed utility anche per C16 su SUPERCOMMODORE. Invitiamo pertanto tutti i lettori che abbiano acquistato il C16 ad inviarci i propri lavori migliori.

Il gioco della Roulette è un classico dell'azzardo. Analizziamo in primo luogo il tipo di giocate che è possibile effettuare con questo programma:

- **NUMERI PIENI**: è la classica giocata su un numero che, se estratto, rende 35 volte la posta.

- **CAVALLI**: gruppi di due numeri, ad esempio 1 e 2 oppure 1 e 4; paga 17 volte la posta.

- **TERZINE**: gruppi di tre numeri consecutivi sulla stessa riga, come 1, 2, 3 oppure 13, 14, 15; paga 11 volte la posta.

- **CARRE**: gruppi di quattro numeri in quadrato, come 1, 2, 4, 5 oppure 14, 15, 17, 18; paga 8 volte la posta.

- **SESTINE**: gruppi di sei numeri consecutivi su due righe, come sestina 1-6 oppure sestina 13-18; paga 5 volte la posta.

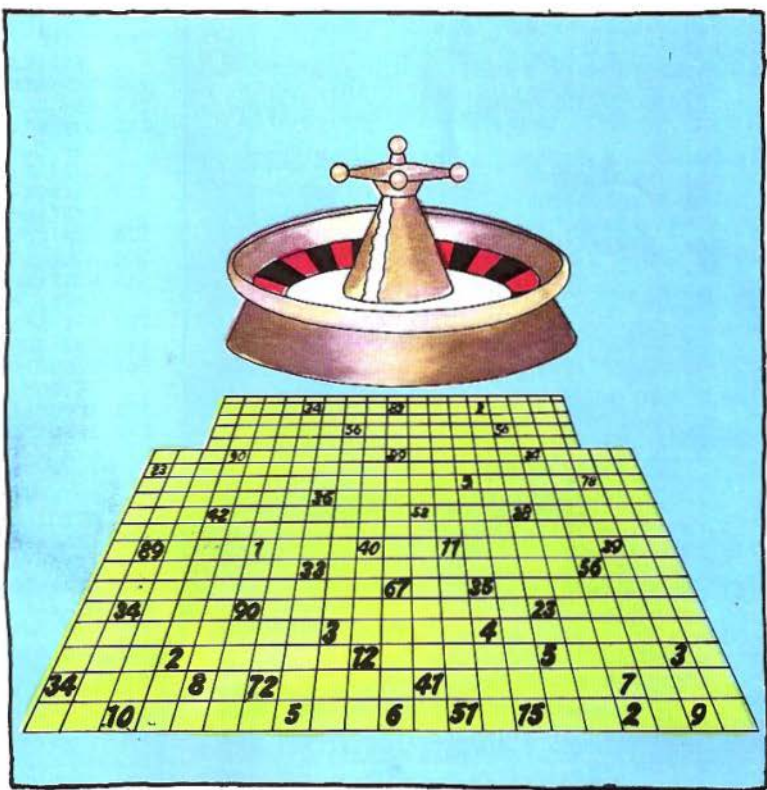
- **DOZZINE**: le tre dozzine giocabili sono la 1-12, la 13-24 e la 25-36; pagano 2 volte la posta.

- **COLONNE**: le tre colonne A, B e C pagano 2 volte la posta.

- **PASSE/MANQUE**: i primi 18 numeri (1-18) sono il Manque, mentre gli ultimi 18 (19-36) sono il Passe; questa giocata paga 1 volta la posta.

- **PARI/DISPARI** e **ROSSO/NERO** pagano 1 volta la posta.

Il programma, dopo essersi informato sul numero dei giocatori (limitato solo



dalla memoria disponibile nel computer: è stato comunque verificato fino a sei giocatori) e sui rispettivi capitali disponibili, passa al primo giocatore e gli chiede quali giocate intenda fare tramite un menu di gioco. Scegliendo la, o le opzioni desiderate, si visualizzerà il tavolo di giocata, ed il programma chiederà numeri e puntata. Per puntate su terzine, carrè, sestine e dozzine sarà sufficiente battere il numero più basso compreso nel gruppo, ed il computer provvederà a calcolare anche gli altri automaticamente. Dopo ogni puntata verrà visualizzato il capitale ancora di-

sponibile e quello giocato; finito di puntare, premendo il tasto di fine giocata si farà in modo che il calcolatore, dopo avervi informati sul totale giocato, passi a raccogliere le puntate del giocatore successivo.

Quando tutti avranno effettuato le proprie giocate verrà estratto il fatidico numero e visualizzati gli esiti, giocatore per giocatore e giocata per giocata. Si passerà quindi a raccogliere le puntate successive con lo stesso sistema sopra descritto.

Il programma è ricco di controlli e quindi, se provate a barare, vi prende sem-

pre in castagna. Se poi malauguratamente finite i vostri soldi, il computer ve ne presterà quanti ne volete, e senza pretenderne la restituzione! Si ricordi che il capitale di inizio va immesso per intero, mentre per tutte le

giocate sarà sufficiente immetterlo in unità di mille lire (così ci si risparmia di digitare tutti gli zeri da post-inflazione!). Volendo partire con un capitale di duecentomila lire si dovrà quindi rispondere alla domanda "Capitale di inizio?" con 200000, mentre per giocare - ad

esempio - diecimila lire sul 7 si dovrà rispondere alla domanda "Numero e puntata?" con 7,10 (oppure 7+RETURN, 10+RETURN).

Non ci resta che augurare a tutti buon divertimento!

Roulette per C16

```

0 REM *****
0 REM *
2 REM * ROULETTE PER C16 *
3 REM *DI M. CRISTUIB GRIZZI *
4 REM * SUPERCOMMODORE *
5 REM *
6 REM *****
7 REM
8 REM
9 REM
10 COLOR 0,1:COLOR4,1:PRINTCHR$(14):VOL8
20 PRINT"{CLR}[BLU][<A>]{ 37 *}[<S>}"
21 FORI=1TO20:PRINT"-TAB(38)"_":NEXT
22 PRINT"{<Z>}{ 37 *}[<X>}"
23 PRINT"{HOME}{ 9 DES}{ 8 GIU' }{RVS}
{WHT}C16 R{SPAZI}O{SPAZI}U{SPAZI}L
{SPAZI}E{SPAZI}T{SPAZI}E{OFF}"}
24 PRINT"{ 2 GIU' }{ 10 DES }{GRN}BY SUPER
CO{ 2 M}ODORE"
25 FORW=1TO2000:NEXT
26 P1=35:P2=17:P3=11:P4=8:P5=5:P6=2:P7=2:
P8=1:P9=1:A$=" VOLTE + LA POSTA"
30 PRINT"{CLR}VUOI LE ISTRUZIONI?"
31 GETW$:IFW$<>"S"ANDW$<>"N"THEN31
32 IFW$="S"THENGOSUB30000
120 COLOR0,1:PRINT"{CLR}{WHT}QUANTI GIOC
ATORI ";
121 INPUTNP:IFNP=0THEN121
124 DIMNM$(45,NP),T$(45,NP),PU$(NP),C(NP)
,ZX$(5):M=1000
125 FORI=1TONP:PRINT"{GIU' }ITAB(3):INPUT
"NOME,PREGO";NO$(GIU)
130 INPUT"CAPITALE DI{SPAZI}INIZIO (>1000
)";A:PU$(I)=A/1000:P$(I)=PU$(I):PRINT
:NEXT:G=1
135 FORPL=1TONP:Z=0:IFPU$(PL)=0ANDC(PL)=0
THEN140
137 GOSUB5000
140 NEXTPL:GOTO550
150 REM SUBR. GIOCATA PIENI****
155 GOSUB10000:PRINT"{GIU' }QUANTI PIENI G
IOCHI?(MAX.09) ";
165 INPUTA$:N=VAL(A$):FORI=0TON-1:PRINTI+
1;
170 INPUT"NUMERO E PUNTATA";NM$(I,PL),A:T
$(I,PL)=A
175 Z=Z+A*M:NEXTI:RETURN
180 REM SUBR.GIOCATA CAVALLI****
185 GOSUB10000:PRINT"{GIU' }QUANTI CAVALLI
GIOCHI?(MAX.06) ";
200 INPUTA$:N=VAL(A$):W=9:FORI=1TON
205 PRINTI;:INPUT"NUMERI E PUNTATA";NM$(W
,PL),NM$(W+1,PL),A:T$(W,PL)=A
206 Z=Z+A*M:W=W+2:NEXT:RETURN
210 REM SUBR.GIOCATA TERZINE****
215 GOSUB10000:PRINT"{GIU' }QUANTE TERZINE
GIOCHI?(MAX.06) ";
225 INPUTA$:N=VAL(A$):FORI=21TON+20
230 PRINTI-20;:INPUT"NUM.PIU'BASSO E PUNT
ATA";NM$(I,PL),A:T$(I,PL)=A
235 REM CONTROLLO INPUT
240 FORW=1TO34STEP3:IFW=NM$(I,PL)THEN250

```

```

245 NEXT:PRINT"{RVS}NUMERO NON CONGRUENTE
{OFF} RIPETERE!":GOSUB1100:GOTO230
250 Z=Z+A*M:NEXTI:RETURN
260 REM SUBR.GIOCATA CARRE'****
265 GOSUB10000:PRINT"{GIU' }QUANTI CARRE'
GIOCHI?(MAX.06) ";
275 INPUTA$:N=VAL(A$):FORI=27TON+26
280 PRINTI-26;:INPUT"NUM.PIU'BASSO E PUNT
ATA";NM$(I,PL),A:T$(I,PL)=A
285 REM CONTROLLO INPUT
290 FORW=3TO36STEP3:IFNM$(I,PL)=WORNM$(I,
PL)=34ORNM$(I,PL)=35THEN299
295 NEXT:GOTO300
299 PRINT"{RVS}NUMERO NON CONGRUENTE{OFF}
RIPETERE!":GOSUB1100:GOTO280
300 Z=Z+A*M:NEXTI:RETURN
310 REM SUBR. GIOCATA SESTINE****
315 GOSUB10000:PRINT"{GIU' }QUANTE SESTINE
GIOCHI?(MAX.06) ";
325 INPUTA$:N=VAL(A$):FORI=33TON+32
330 PRINTI-32;:INPUT"NUM.PIU'BASSO E PUNT
ATA";NM$(I,PL),A:T$(I,PL)=A
335 REM CONTROLLO INPUT
340 FORW=1TO34STEP3:IFW=NM$(I,PL)THEN350
345 NEXT:PRINT"{RVS}NUMERO NON CONGRUENTE
{OFF} RIPETERE!":GOSUB1100:GOTO330
350 Z=Z+A*M:NEXTI:RETURN
360 REM SUBR.GIOCATA COLONNE****
365 GOSUB10000:PRINT"{GIU' }QUANTE COLONNE
GIOCHI?(MAX.2) ";
375 INPUTA$:N=VAL(A$):FORI=41TON+40
380 INPUT"COLONNA E PUNTATA";A$,A:T$(I,PL)
)=A
381 IFA$<>"A"ANDA$<>"B"ANDA$<>"C"THENPRIN
T"{RVS}NON CONGRUENTE{OFF} RIPETERE":
GOTO380
385 IFA$="A" THENNM$(I,PL)=1
390 IFA$="B"THENNM$(I,PL)=2
395 IFA$="C"THENNM$(I,PL)=3
405 Z=Z+A*M:NEXTI:RETURN
410 REM SUBR.GIOCATA DOZZINE****
415 GOSUB10000:PRINT"{GIU' }QUANTE DOZZINE
?(MAX.2) ";
425 INPUTA$:N=VAL(A$):FORI=39TON+38
430 INPUT"DOZZINA E PUNTATA";NM$(I,PL),A:
T$(I,PL)=A
435 REM CONTROLLO INPUT
440 IFNM$(I,PL)<>1ANDNM$(I,PL)<>2ANDNM$(I
,PL)<>3THENPRINT"{RVS}NON CONGRUENTE
{OFF}":GOTO430
450 Z=Z+A*M:NEXTI:RETURN
460 GOSUB10000:REM GIOCATA PASSE-MANQUE
465 INPUT"{GIU' }PASSE O MANQUE?(P/M) E PU
NTATA";A$,A:T$(43,PL)=A
466 IFA$<>"P"ANDA$<>"M"THENPRINT"{RVS}NON
CONGRUENTE{OFF} RIPETERE":GOTO465
467 Z=Z+A*M
475 IFA$="P"THENNM$(43,PL)=1:RETURN
480 NM$(43,PL)=2:RETURN
490 GOSUB10000:REM SUBR.GIOCATA PARI-DISP
ARI E COLORE
495 INPUT"{GIU' }P/D O R/N ,PUNTATA";A$,W
497 Z=Z+W*M

```



```

505 IFA$="P"THENNM%(44,PL)=1:T%(44,PL)=W:
RETURN
506 IFA$="D"THENNM%(44,PL)=2:T%(44,PL)=W:
RETURN
507 IFA$="R"THENNM%(45,PL)=1:T%(45,PL)=W:
RETURN
508 IFA$="N"THENNM%(45,PL)=2:T%(45,PL)=W:
RETURN
510 PRINT"[RVS]NON CONGRUENTE{OFF} RIPETE
RE":Z=Z-W*M:GOTO495
550 REM ESTRAZIONE E CLASSIFICAZIONE
551 IFZX>4THENZX=0
559 X=INT(RND(TI)*37):ZX%(ZX)=X:ZX=ZX+1
560 IFX<19THENPMS="MANQUE":GOTO562
561 PMS="PA{ 2 S}E"
562 IFX/2=INT(X/2)THENPDS="PARI":GOTO564
563 PDS="DISPARI"
564 FORI=1TO3STEP2:IFX=ITHENRNS="NERO":G
OTO566
565 NEXT:RNS="RO{ 2 S}O"
566 FORI=1TO3STEP3:IFX=ITHENCOS="A":GOTO
569
567 II=I+1:IFX=ITHENCOS="B":GOTO569
568 NEXT:COS="C"
569 IFX=0THENPDS="":RNS="":COS="":PMS=""
570 REM VISUALIZZAZIONE****
576 FORI=1TO10:PRINT"{CLR}{ 11 GIU' }
{ 12 DES}{RVS}{YEL}A{ 2 T}ENDERE PRE
GO{OFF}{<8>":FORW=1TO100:NEXT
577 PRINT"{CLR}":SOUND 2,800,2:FORW=1TO10
0:NEXTW,I
580 FORPL=1TONP:PRINT"{CLR}":IFPU%(PL)=0A
NDC(PL)=0THEN1085
585 PRINT"{YEL}*****
USCITO IL {RVS}{<5>}"X{<8>}"
586 PRINTPDS;" ";RNS;" ";PMS;" COL. ";COS
:PRINT
590 PRINT"RISULTATI{SPAZI}DI {RVS}{<7>}"N
OS(PL)"{GIU' }{<8>}"
595 REM PIENI****
600 FORW=0TO8:IFT%(W,PL)=0THEN650
605 IFX=NM%(W,PL)THEN620
610 PRINT"PIENO"TAB(10)NM%(W,PL)TAB(30)"P
ERDE":PU%(PL)=PU%(PL)-T%(W,PL):GOTO62
5
620 PRINT"PIENO"TAB(10)NM%(W,PL)TAB(30)"*
VINCE!":PU%(PL)=PU%(PL)+P1*T%(W,PL)
625 NEXTW
650 REM CAVALLI****
655 FORW=9TO20STEP2:IFT%(W,PL)=0THEN700

```

```

660 IFX=NM%(W,PL)ORX=NM%(W+1,PL)THEN675
665 PRINT"CAVALLO"TAB(10)NM%(W,PL)"-NM%(
W+1,PL)TAB(30)"PERDE"
670 PU%(PL)=PU%(PL)-T%(W,PL):GOTO690
675 PRINT"CAVALLO"TAB(10)NM%(W,PL)"-NM%(
W+1,PL)TAB(30)"*VINCE"
680 PU%(PL)=PU%(PL)+P2*T%(W,PL)
690 NEXTW
700 REM TERZINE****
705 FORW=21TO26:IFT%(W,PL)=0THEN750
710 IFX=NM%(W,PL)ORX=NM%(W,PL)+1ORX=NM%(
W,PL)+2THEN725
715 PRINT"TERZINA"TAB(10)NM%(W,PL)NM%(W,P
L)+1NM%(W,PL)+2TAB(30)"PERDE"
720 PU%(PL)=PU%(PL)-T%(W,PL):GOTO735
725 PRINT"TERZINA"TAB(10)NM%(W,PL)NM%(W,P
L)+1NM%(W,PL)+2TAB(30)"*VINCE"
730 PU%(PL)=PU%(PL)+P3*T%(W,PL)
735 NEXTW
750 REM CARRE'****
755 FORW=27TO32:IFT%(W,PL)=0THEN790
760 A=NM%(W,PL):IFX=AORX=A+1ORX=A+3ORX=A+
4THEN775
765 PRINT"CARRE'"TAB(10)A;A+1;A+3;A+4TAB(
30)"PERDE"
770 PU%(PL)=PU%(PL)-T%(W,PL):GOTO785
775 PRINT"CARRE'"TAB(10)A;A+1;A+3;A+4TAB(
30)"*VINCE"
780 PU%(PL)=PU%(PL)+P4*T%(W,PL)
785 NEXTW
790 REM SESTINE****
795 FORW=33TO38:IFT%(W,PL)=0THEN850
800 FORI=NM%(W,PL)TONM%(W,PL)+5:IFX=ITHEN
815
805 NEXTI:A=NM%(W,PL):PRINT"SESTINA"TAB(1
0)A-"A+5TAB(30)"PERDE"
810 PU%(PL)=PU%(PL)-T%(W,PL):GOTO825
815 PRINT"SESTINA"TAB(10)A-"A+5TAB(30)"*
VINCE"
820 PU%(PL)=PU%(PL)+P5*T%(W,PL)
825 NEXTW
850 REM COLONNE****
855 FORW=41TO42:IFT%(W,PL)=0THEN880
856 IFNM%(W,PL)=1THENAS="A"
857 IFNM%(W,PL)=2THENAS="B"
858 IFNM%(W,PL)=3THENAS="C"
860 IFCOS=ASTHEN870
865 PRINT"COLONNA"TAB(11)ASTAB(30)"PERDE"
:PU%(PL)=PU%(PL)-T%(W,PL):GOTO875
870 PRINT"COLONNA"TAB(11)ASTAB(30)"*VINC
E":PU%(PL)=PU%(PL)+P7*T%(W,PL)
875 NEXTW
880 REM DOZZINE****
885 FORW=39TO40:IFT%(W,PL)=0THEN950
890 IFX>0ANDX<13ANDNM%(W,PL)=1THEN910
895 IFX>12ANDX<25ANDNM%(W,PL)=2THEN910
900 IFX>24ANDNM%(W,PL)=3THEN910
905 PRINT"DOZZINA"TAB(10)NM%(W,PL)TAB(30)
"PERDE":PU%(PL)=PU%(PL)-T%(W,PL)
906 GOTO920
910 PRINT"DOZZINA"TAB(10)NM%(W,PL)TAB(30)
"*VINCE"
915 PU%(PL)=PU%(PL)+P6*T%(W,PL)
920 NEXTW
950 REM PASSE-MANQUE****
955 IFT%(43,PL)=0THEN990
960 IFNM%(43,PL)=1THENAS="PA{ 2 S}E"
965 IFNM%(43,PL)=2THENAS="MANQUE"
970 IFPMS=ASTHEN980
975 PRINTASTAB(30)"PERDE":PU%(PL)=PU%(PL)
-T%(43,PL):GOTO990

```

GIOCATA n. 3 DI

premi 1	per numeri pieni
premi 2	per cavalli
premi 3	per terzine
premi 4	per carre'
premi 5	per sestine
premi 6	per dozzine
premi 7	per colonne
premi 8	per passe/manque
premi 9	per p/d e r/n
premi 10	per fine giocata

TOT. POSSIBILI ATTUALMENTE 1228888
DISPONIBILI DA GIOCARE 1178888
TOTALE GIOCATE 50000

ultimi 5 numeri usciti:
 0 25 0 8 0

LOAD


**Roulette
per C16**

```

980 PRINTASTAB (30) "*VINCE": PU%(PL) = PU%(PL)
    )+T%(43, PL)
990 REM PARI DISPARI****
995 IFT%(44, PL) = 0 THEN 1025
1000 IFNM%(44, PL) = 1 THEN AS = "PARI"
1005 IFNM%(44, PL) = 2 THEN AS = "DISPARI"
1010 IFPDS = AS THEN 1020
1015 PRINTASTAB (30) "PERDE": PU%(PL) = PU%(PL)
    )-T%(44, PL): GOTO 1025
1020 PRINTASTAB (30) "*VINCE": PU%(PL) = PU%(PL)
    )+T%(44, PL)
1025 REM COLORE****
1029 IFT%(45, PL) = 0 THEN 1060
1030 IFNM%(45, PL) = 1 THEN AS = "RO [ 2 S] O"
1035 IFNM%(45, PL) = 2 THEN AS = "NERO"
1040 IFRNS = AS THEN 1050
1045 PRINTASTAB (30) "PERDE": PU%(PL) = PU%(PL)
    )-T%(45, PL): GOTO 1060
1050 PRINTASTAB (30) "*VINCE": PU%(PL) = PU%(PL)
    )+T%(45, PL)
1060 REM ESITO GIOCATA****
1065 E = PU%(PL) - P%(PL): ES = STR$(E*1000): IFE
    > 0 THEN ES = "+" + RIGHT$(ES, LEN(ES) - 1)
1066 PRINT "{GIU'} ESITO GIOCATA: " ES
1070 PRINT "NUOVO { SPAZI } CAPITALE: " PU%(PL) *
    1000
1071 IFPU%(PL) > 0 THEN 1075
1072 INPUT "VUOI CONTINUARE (S/N) "; RS: IFR
    $ = "N" THEN PU%(PL) = 0: C(PL) = 0: GOTO 1085
1073 INPUT "QUANTO IN CREDITO"; W: PU%(PL) = W
    /M: C(PL) = C(PL) + W: GOTO 1085
1075 PRINT "{ 25 DES } {RVS} PREMI UN TASTO
    {OFF}"
1080 GETAS: IFAS = "" THEN 1080
1085 PRINT "{CLR}": NEXT PL
1090 G = G + 1: GOSUB 20000: GOTO 135
1100 FOR W = 1 TO 1000: NEXT: RETURN
5000 PRINT "{CLR}": GOSUB 6000: IFPU%(PL) > C(P
    L) / MANDC(PL) > 0 THEN PU%(PL) = PU%(PL) - C(
    PL) / M: C(PL) = 0
5001 COLOR 0, 1: PRINT "{HOME} {WHT} GIOCATA N
    " "G" DI {RVS} [<7>] "NO$(PL) "{OFF} [<8>]
5002 PRINT "{GIU'} PREMI {RVS} [<7>] 1 {OFF}
    [<8>] PER NUMERI PIENI"
5003 PRINT "PREMI {RVS} [<7>] 2 {OFF} [<8>] PE
    R CAVALLI": PRINT "PREMI {RVS} [<7>] 3
    {OFF} [<8>] PER TERZINE"
5004 PRINT "PREMI {RVS} [<7>] 4 {OFF} [<8>] PE
    R CARRE": PRINT "PREMI {RVS} [<7>] 5
    {OFF} [<8>] PER SESTINE"
5005 PRINT "PREMI {RVS} [<7>] 6 {OFF} [<8>] PE
    R DOZZINE": PRINT "PREMI {RVS} [<7>] 7
    {OFF} [<8>] PER COLONNE"
5006 PRINT "PREMI {RVS} [<7>] 8 {OFF} [<8>] PE
    R PASSE/MANQUE": PRINT "PREMI {RVS}
    [<7>] 9 {OFF} [<8>] PER P/D E R/N"
5007 PRINT "PREMI {RVS} [<7>] 0 {OFF} [<8>] PE
    R FINE GIOCATA"
5008 PRINT "{GIU'} TOT.PO { 2 S} EDUTO A
    { 2 T} UALMENTE { 2 DES} " PU%(PL) * 1000
5009 IFPU%(PL) > 0 THEN PRINT "DISPONIBILE D
    A GIOCARE { 5 DES} " PU%(PL) * 1000 - Z
5010 IF (PU%(PL) * 1000 - Z) > 0 THEN 5013
5011 PRINT "{RVS} GIOCATA { SPAZI } SUPERIORE
    { SPAZI } A { SPAZI } QUANTO { SPAZI } PO
    { 2 S} EDUTO { SPAZI } E PERTANTO { SPAZI }
    NON { SPAZI } A { 2 C} E { 2 T} ATA"
5012 FOR I = 1 TO 2000: NEXT: Z = 0: GOTO 5000
5013 IF C(PL) > 0 THEN PRINT "{RVS} {YEL} DEBITO
    " TAB (27) C(PL) " [<8>] "
5019 GETWS: IFWS = "" THEN 5019
5020 IFWS = "{F1}" THEN PRINT "{CLR}": END
5021 IFWS = "0" THEN PRINT "TOTALE GIOCATO
    { 13 DES} {GRN} " Z " [<8>] ": FOR I = 1 TO 1000
    : NEXT: RETURN
5022 ON VAL (WS) GOSUB 150, 180, 210, 260, 310, 41
    0, 360, 460, 490
5023 GOTO 5000
6000 REM NUMERI USCITI
6005 PRINT "{ 21 GIU' } ULTIMI 5 NUMERI USCI
    TI: "
6010 FOR I = 0 TO 4: PRINT ZX%(I); : NEXT: PRINT "
    {HOME}": RETURN
6011 FOR I = 0 TO 4: PRINT ZX%(I); : NEXT: PRINT "
    {HOME}": RETURN
6020 FOR I = 0 TO 4: PRINT ZX%(I); : NEXT: PRINT "
    {HOME}": RETURN
10000 COLOR 0, 2: REM SUBROUTINE TAVOLO VER
    DE
10001 PRINT "{CLR} { 2 SPAZI } {GRN} [<A>]
    { 14 * } [<R>] { 17 * } [<S>]"
10002 PRINT "{ 2 SPAZI } {GRN} [<8>]
    { 2 SPAZI } M A N Q U E {GRN} [<8>]
    { 2 SPAZI } P A S S E { 6 SPAZI } {GRN}
    -"
10003 PRINT "{GRN} { 2 SPAZI } [<Q>] { 8 * }
    [<R>] { 5 * } [<E>] { 5 * } [<R>]
    { 11 * } [<W>]"
10004 PRINT "{GRN} { 2 SPAZI } [<8>] PREMIERE
    {GRN} - [<8>] DEUXIEME { 2 SPAZI } {GRN}
    - [<8>] TROISIEME {GRN} -"
10005 PRINT "{GRN} [<A>] * + [<R>] * [<R>] *
    [<R>] { 2 * } + { 2 * } [<R>] { 2 * } [<R>]
    { 2 * } [<R>] { 2 * } + { 2 * } [<R>]
    { 2 * } [<R>] { 2 * } [<R>] { 2 * } * + [<S>]"
10006 PRINT "{GRN} - {RED} 3 {GRN} - {BLK} 6
    {GRN} - {RED} 9 {GRN} - {BLK} 12 {GRN} -
    {RED} 15 {GRN} - {BLK} 18 {GRN} - {RED} 21
    {GRN} - {BLK} 24 {GRN} - {RED} 27 {GRN} -
    {BLK} 30 {GRN} - {RED} 33 {GRN} - {BLK} 36
    {GRN} - [<1>] C {GRN} -"
10007 PRINT "{GRN} - [<Q>] * + * + * + { 2 * } +
    { 2 * } + { 2 * } + { 2 * } + { 2 * } +
    { 2 * } + { 2 * } + { 2 * } + { 2 * } * + [<W>]"
10008 PRINT "{GRN} - [<1>] 0 {GRN} - {BLK} 2 {GRN}
    - {RED} 5 {GRN} - {BLK} 8 {GRN} - {RED} 11
    {GRN} - {BLK} 14 {GRN} - {RED} 17 {GRN} -
    {BLK} 20 {GRN} - {RED} 23 {GRN} - {BLK} 26
    {GRN} - {RED} 29 {GRN} - {BLK} 32 {GRN} -
    {RED} 35 {GRN} - [<1>] B {GRN} -"
10009 PRINT "{GRN} - [<Q>] * + * + * + { 2 * } +
    { 2 * } + { 2 * } + { 2 * } + { 2 * } +
    { 2 * } + { 2 * } + { 2 * } + { 2 * } * + [<W>]"
10010 PRINT "{GRN} - {RED} 1 {GRN} - {BLK} 4
    {GRN} - {RED} 7 {GRN} - {BLK} 10 {GRN} -
    {RED} 13 {GRN} - {BLK} 16 {GRN} - {RED} 19
    {GRN} - {BLK} 22 {GRN} - {RED} 25 {GRN} -
    {BLK} 28 {GRN} - {RED} 31 {GRN} - {BLK} 34
    {GRN} - [<1>] A {GRN} -"
10011 PRINT "{GRN} [<Z>] * [<E>] * [<E>] * [<E>]
    * [<E>] { 2 * } [<E>] { 2 * } [<E>]
    { 2 * } [<E>] { 2 * } [<E>] { 2 * } [<E>]
    { 2 * } [<E>] { 2 * } [<E>] { 2 * } [<E>]
    { 2 * } [<E>] * [<X>] [<1>": RETURN

```

```

20000 REM SUBR.INIZIALIZZAZ.VARIABILI
20005 FORI=1TONP:P%(I)=PU%(I):NEXT
20015 FORW=0TO45:FORI=1TONP:NM%(W,I)=0:T%(
(W,I)=0:NEXT:NEXT
20020 W=0:N=0:A=0:AS="":PS="":CS="":NS="":
:M$=""
20025 RETURN
30000 COLOR0,2:PRINT"{GRN}";
30010 PRINT"{CLR}{RVS}ISTRUZIONI{SPAZI}D
I{SPAZI}GIOCO"
30020 PRINT"{ 2 GIU' }TUTTE LE PUNTATE VAN
NO IMMESSE IN UNITA'"
30030 PRINT"DI MILLE LIRE,TRANNE IL CAPIT
ALE DI INI-"
30040 PRINT"ZIO E GLI EVENTUALI CREDITI D
EL BANCO."
30050 PRINT"{GIU'}PER USCIRE DAL PROGRAMM
A PREMI F1 ALLA"
30060 PRINT"{GIU'}VISUALIZZAZIONE DEL MEN
U DI GIOCO."
30070 PRINT"{ 3 GIU' }PREMI UN TASTO"
30080 GETKEYS
30090 COLOR 0,1:PRINT"{CLR}{WHT}{RVS}
{ 6 GIU' } { 5 DES }QUOTE{SPAZI}DI
{SPAZI}PAGAMENTO"
30091 PRINT"{GIU'}{CYN}**** PIENI"TAB(18)
P1;A$:PRINT"{GIU'}**** CAVA{ 2 L}
I"TAB(18)P2;A$
30092 PRINT"{GIU'}**** TERZINE"TAB(18)P3;
A$
30093 PRINT"{GIU'}**** CA{ 2 R}E"TAB(19)
P4;A$:PRINT"{GIU'}**** SESTINE"TAB(
19)P5;A$

```

MANQUE			DEUXIEME			TROISIEME							
3	6	9	12	15	18	21	24	27	30	33	36	e	
6	2	5	8	11	14	17	20	23	26	29	32	35	b
1	4	7	10	13	16	19	22	25	28	31	34	a	

quanti pieni giochi? (max. 69) ? 3
I numero e puntata? 1.18
W numero e puntata? 3.48
M numero e puntata? 13.58

```

30094 PRINT"{GIU'}**** DO{ 2 Z}INE"TAB(19)
)P6;A$
30095 PRINT"{GIU'}**** COLO{ 2 N}E"TAB(19)
)P7;A$:PRINT"{GIU'}**** PA{ 2 S}E/
MANQUE"TAB(19)P8;A$
30096 PRINT"{GIU'}**** COLORE"TAB(19)P9;A
$
30097 PRINT"{GIU'}**** PARI/DISPARI"TAB(1
9)P9;A$
30098 PRINT"{GIU'}{ 9 DES}PREMI{SPAZI}UN
{SPAZI}TASTO PER INIZIARE"
30099 GETKEYA$
30100 COLOR 0,1:RETURN

```

IN EDICOLA

L. 2.000
L. 2.000
Rivista di Informatica Pratica - gruppo 122/79

COMPUSCUOLA

La rivista di informatica nella didattica per la scuola italiana

Jean Jacques Rousseau...

"Jean Jacques Rousseau, il suo tempo maturo in guardia i genitori dai rischi, portare le idee ai bambini, poi il grande acculturato, parlo del mio tempo fu Saigon, con le sue avventure fantasche, poi i turisti, per la salita di montagna, poi i mezzi

autonomi, la televisione in casa, ma anche i PC. E allora se la prendono con i computer, la storia, questo storia, si parte con gli "storie argomentati". A parlare è Maurizio Laing, intervistato da Compuscuola e fluita alla sede del Magistero. Dispari come vanno attualmente le cose?

"Siamo tra le due pinze di una famiglia. Siamo antiche, ma il primo secolo, quasi nebre il primo secolo, quasi di natura fisica, oggettivo. Negli anni '80 ci si battono perché si accendessero in modo intelligente gli studenti del tutto coperto di un accio e passavano dai mezzi e inesplicitamente costrui in cui i ragazzi erano costretti a scuola, costretti di prodotti elettronici da altri. Questo avveniva con un unco verso finale: l'accettazione di questi strumenti di "acchi che diventavano" non è più formulazione ma, le si può trovare nell'autorevolissimo rapporto di Federico e Maurizio. Buone bene che non non avveniva solo in

«Vota i migliori Personal Computer»

Tra le tante iniziative editoriali che hanno fatto da cornice al Bias troviamo anche un'indagine: "Vota i migliori Personal Computer". L'indagine, condotta dall'Istituto Makno, è stata promossa dalla Segreteria di Informatica del Gruppo Editoriale Jackson con la collaborazione di Alphard, IM Italia, Vir Video Technology Research, Cofedito dell'iniziativa era di rilevare le preferenze espresse dai visitatori in merito a una classifica dei migliori personal computer suddivisi nelle categorie home (prezzo base inferiore a 1 milione) e professional (prezzo base superiore a 1 milione) e adatti per la loro caratteristiche e funzionali, si

Barbara
Nabiz e sviluppo, annessi e isolati (pag. 2)
Elle computer per il recupero dei dati
Gianni Vastar intervista sul tema (pag. 7)
Software
Abbiamo visitato per voi (pag. 8)
Libri (pag. 12)
Comandare con il calcolatore
Fortino, Basic, Pascal, con Logo, Prolog... (1)

Una pubblicazione firmata...



GRUPPO EDITORIALE JACKSON

20124 MILANO - Via Rosellini, 12
Tel. 02/6880451/2/3/4/5

Novità Jackson.

David Lawrence
**LINGUAGGIO
 MACCHINA DEL
 COMMODORE 64**

Il libro apre nuovi orizzonti a tutti coloro che sono interessati alla programmazione in linguaggio macchina del COMMODORE 64.

Con cassetta
 Cod. 572D Pag. 208 Lire 29.000

Clive Prigmore
**IL BASIC IN 30 ORE
 PER SPECTRUM**

Questo semplice corso di autoistruzione insegna a programmare, e un programma ha sempre bisogno di due ingredienti, un linguaggio e una struttura: dunque questo libro non insegna solo il BASIC, ma anche come si organizza correttamente un buon programma.

Cod. 501B Pag. 360 Lire 40.000

Rodnay Zaks
**IL TUO PRIMO
 PROGRAMMA IN BASIC**

La diffusione del BASIC per la sua semplicità e quasi "naturalità" di programmazione fa sì che una cultura generale sull'informatica e la sua applicazione non può prescindere da una conoscenza di base di questo linguaggio. Questo lo scopo del libro: permettere anche a chi ha soltanto una cultura di base, di capire che cos'è il BASIC e come si usa.

Cod. 507B Pag. 216 Lire 19.500

Czes Kosniowski
**MATEMATICA E
 COMMODORE 64**

Tutte le funzioni matematiche disponibili sul C64 sono qui descritte, ed il loro uso è illustrato con programmi che possono essere utilizzati dal lettore all'interno dei suoi, per particolari applicazioni.

Il libro contiene anche informazioni e programmi su altri argomenti, come i codici e la crittografia, i numeri casuali, le serie, la trigonometria, i numeri primi e l'analisi statistica dei dati.

Con cassetta
 Cod. 570D Pag. 160 Lire 24.000

Williams
**AI CONFINI
 DELLO SPECTRUM
 Applicazioni avanzate**

Un esame attento dei listati consentirà al lettore di apprendere i "segreti" della programmazione strutturata e migliorare notevolmente le proprie capacità di programmatore. I programmi presentati vanno dagli arcade più famosi, tra cui il celebre "Spectrum Invaders" ai programmi di utilità più interessanti, dai giochi d'azzardo ai programmi didattici, dai programmi funzionali a quelli di matematica e di giochi di strategia.

Con cassetta
 Cod. 414B Pag. 180 Lire 28.000

Mike Grace
**ADVENTURE E
 COMMODORE 64**

Un manuale per ideare e utilizzare programmi di Adventure basati esclusivamente sul testo.

Una progettazione modulare del programma rende più facile la comprensione della struttura. L'abilità nella programmazione avrà modo di emergere durante la costruzione graduale dell'Adventure, per modificare il programma dimostrativo, o addirittura per costruirne uno nuovo.

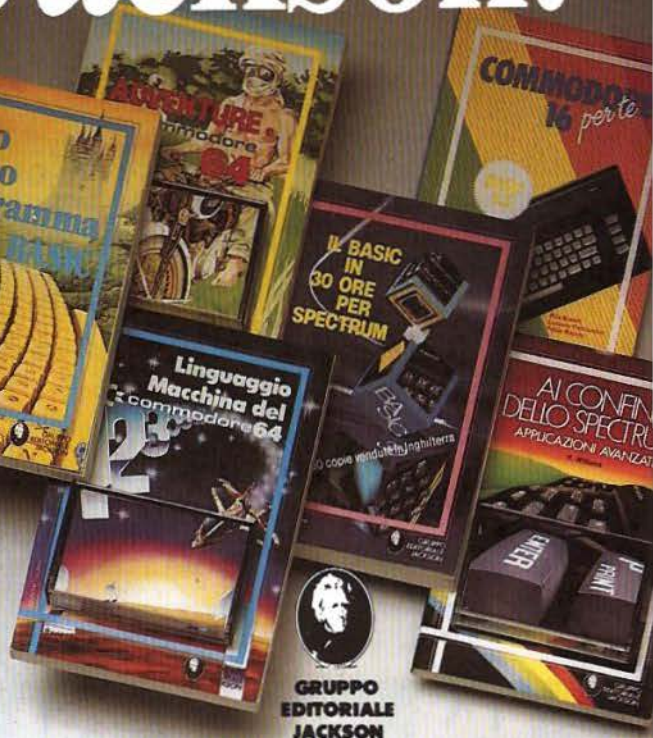
Con cassetta
 Cod. 571D Pag. 240 Lire 35.000

Rita Bonelli
**COMMODORE 16
 PER TE: BASIC 3.5**

È un libro di introduzione al BASIC C16 con il classico taglio didattico Bonelli. Adatto per la Scuola media inferiore e per chi non conosce l'informatica.

La cassetta allegata al libro contiene diverse lezioni, una per ogni capitolo, che devono essere lette prima del capitolo relativo.

Con cassetta
 Cod. 413B Pag. 296 Lire 35.000



**GRUPPO
 EDITORIALE
 JACKSON**

La biblioteca che fa testo.

ritagliare (o fotocopiare) e spedire in busta chiusa a:
GRUPPO EDITORIALE JACKSON - Divisione Libri - Via Rosellini, 12 - 20124 Milano
CEDOLA DI COMMISSIONE LIBRARIA

VOGLIATE SPEDIRMI

n° copie	codice	Prezzo unitario	Prezzo totale
Totale			

Pagherò contrassegno al postino il prezzo indicato più L. 3.000 per contributo fisso spese di spedizione.

Condizioni di pagamento con esenzione del contributo spese di spedizione:

- Allego assegno della Banca Allego fotocopia del versamento su c/c n. 11666203 a voi intestato
- n° _____ Allego fotocopia di versamento su vaglia postale a voi intestato

Nome _____
 Cognome _____
 Via _____
 Cap _____ Città _____ Prov. _____
 Data _____ Firma _____

Spazio riservato alle Aziende. Si richiede l'emissione di fattura **ORDINE MINIMO L. 50.000**

Partita I.V.A. _____



Magic Draw per Vic 20

di **K. Gough**
trad. ed adatt.
di **M. Cristuib Grizzi**

Con questo programma potrete trasformare il vostro computer in una tavoletta grafica ad alta risoluzione. I disegni generati possono essere salvati su nastro o disco, ed è richiesta un'espansione di memoria di almeno 8 Kbyte.

Ecce un programma che permette di creare disegni in alta risoluzione su una griglia di 176x160 pixel. Vengono inserite in memoria tre routine in linguaggio macchina, con la funzione di salvare e caricare in memoria i disegni in alta risoluzione che avete creato, e di pulire la pagina grafica.

Queste routine sono contenute nelle istruzioni DATA del programma e richiedono, se non disponete della cassetta allegata, particolare attenzione nella battitura, in quanto un errore in queste istruzioni può portare con alta probabilità al blocco totale del computer.

Ecco ora un elenco dei comandi disponibili con "Magic Draw":

SHIFT/CLR

Pulisce lo schermo in alta risoluzione. Dopo aver attivato "Magic Draw" per la prima volta, dovrete usare questo comando per pulire lo schermo in alta risoluzione dai disegni casuali che saranno in esso contenuti. Dopo aver premuto SHIFT/CLR potrete vedere il pixel lampeggiante (cursore) al centro dello schermo.

SHIFT/CRSR (SU)

Muove il cursore verso l'alto di un pixel.

CRSR (GIÙ)

Muove il cursore verso il basso di un pixel.

SHIFT/CRSR (SINISTRA)

Muove il cursore verso sinistra di un pixel.

CRSR (DESTRA)

Muove il cursore verso destra di un pixel.

HOME

Porta il cursore al centro dello schermo.

X

Plotta un pixel alla posizione corrente del cursore.

Z

Cancella - se presente - un pixel alla posizione del cursore.

D Plotta in modo continuo, muovendo il cursore con i tasti CRSR. Premete ancora D per disabilitare questa funzione.

E Cancella in modo continuo, muovendo il cursore con i tasti CRSR. Premete ancora D per disabilitare questa funzione.

@

Il tasto "@" crea una figura a rettangolo pieno a destra e verso il basso rispetto al cursore. Le dimensioni del rettangolo sono controllate dai tasti "+" e "-".

+

Incrementa le dimensioni del rettangolo di un pixel ogni volta che sia premuto questo tasto. Il limite massimo è di 90 pixel.

-

Decrementa le dimensioni del rettangolo di un pixel ogni volta che sia premuto questo tasto. Il minimo è di 2 pixel.

0

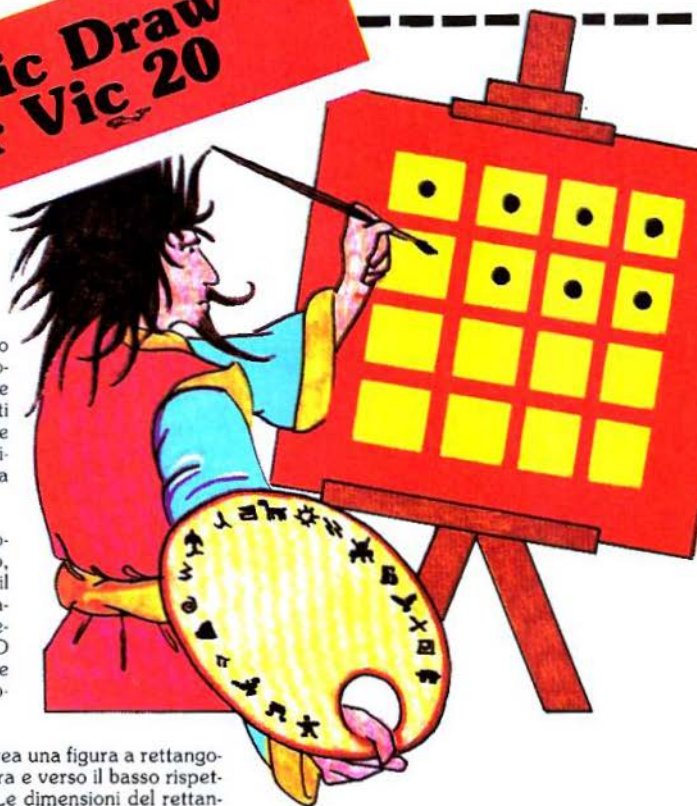
Il tasto "0" crea una figura a rettangolo nel colore del fondo dello schermo, a destra e verso il basso rispetto alla posizione del cursore. Le dimensioni del rettangolo sono controllate dai tasti "+" e "-". Questa funzione è utilizzabile per aprire "finestre" su zone dello schermo completamente nere.

G

Disegna sullo schermo una griglia di spazi-carattere (8x8 pixel), utile per tracciare disegni ad elevata precisione. Premete SHIFT/G per cancellare la griglia.

S

Salva un disegno su nastro o disco: ogni pagina grafica occupa 3524 byte. La pressione di S provoca l'uscita dalla pagina grafica e la richiesta del nome del file. Quindi si dovrà digitare 1 per disco o 2 per il nastro. Una volta completato



il salvataggio, si ritorna nella pagina grafica. Gli utilizzatori di unità a cassetta dovranno assicurarsi che i tasti sul registratore siano premuti. Verrà domandato "Tape off?" prima della comparsa della scritta "Press Record & Play on Tape". Premete semplicemente RETURN alla domanda "Tape off?", e quindi premete i tasti Record e Play sul registratore.

L

Carica un disegno da nastro o disco. Si dovrà digitare 1 per il disco o 2 per il nastro, ed il nome del file. Una volta completato il caricamento si entra automaticamente in modo "D" ed il disegno viene visualizzato sullo schermo con il cursore al centro di quest'ultimo.

Q

Esce dal modo in alta risoluzione: lo schermo si pulisce e ricompare il normale cursore del VIC 20. È possibile dare nuovamente il RUN al programma e rivisualizzare il disegno precedentemente tracciato.

F2

Premendo il tasto di funzione F2 viene stampato il disegno in alta risoluzione su una stampante Commodore VIC 1515 o 1525. La stampa dell'intero schermo richiede circa 35 minuti.

Magic Draw per VIC 20

programma 1

```
20 POKE648,30:SYS58648:REM MUOVE LO SCHER
MO A 7680 :rem 118
25 REM :rem 75
30 PRINT"{CLR}{, 4 GIU'}LOAD{ 3 SU}"
:rem 222
40 POKE 631,13:POKE198,1:REM RETURN AUTO
MATICO :rem 25
50 POKE8192,0:POKE43,1:POKE44,32:NEW:REM
MUOVE IL BASIC A 8192 :rem 45
```

Magic Draw per VIC 20

programma 2

```
50 IF PEEK(648)<>30THENPRINT"{CLR}{GIU'}N
ON E' STATO MANDATO IN ESECUZ.IL PROGR
.!!!!":END :rem 249
60 PRINT"{CLR}":POKE36878,15:GOSUB970
:rem 47
90 POKE36867,149:POKE36869,252:POKE36879,
233 :rem 236
100 FORI=0TO7:P%(I)=2E(7-I):NEXT:rem 149
120 FORI=0TO219:POKE7680+I,I:POKE38400+I,
0:NEXTI:X=88:Y=80 :rem 7
200 GETA$:POKE36876,0:IFA$="{SU}"THENY=Y-
1:GOTO900 :rem 114
220 IFA$="{GIU'}"THENY=Y+1:GOTO900
:rem 160
230 IFA$="{SIN}"THENX=X-1:GOTO900
:rem 45
240 IFA$="{DES}"THENX=X+1:GOTO900
:rem 172
250 IFA$="D"THENCW%=NOTCW%:POKE36876,220:
GOSUB500:GOTO300 :rem 183
260 IFA$="E"THENEW%=NOTEW%:POKE36876,158:
GOSUB500 :rem 193
280 IFA$=CHR$(147)THENGOSUB980:SYS319:A$=
CHR$(19) :rem 119
290 IFA$=CHR$(19)THENX=88:Y=80:POKE36876,
200:GOSUB500 :rem 120
300 IFA$="S"THENGOSUB550:GOSUB560:POKE251
,LEN(N$):SYS675:SYS694:ON-(S=1)GOSUB9
96:RUN :rem 236
310 IFA$="L"THENGOSUB550:GOSUB560:POKE251
,LEN(N$):SYS675:SYS712:ON-(S=1)GOSUB9
96:RUN :rem 221
320 IFA$="Q"THENGOSUB550:END :rem 181
330 IFA$="G"THENA$="X":GOTO410 :rem 199
340 IFA$=CHR$(199)THENA$="Z":GOTO410
:rem 52
350 IFA$="+"THENQ%=Q%+1:POKE36876,160+Q%:
IFQ%>90THENQ%=90 :rem 248
360 IFA$="-"THENQ%=Q%-1:POKE36876,250-Q%:
IFQ%<1THENQ%=1 :rem 141
370 IFA$="@"THENA$="X":GOTO430 :rem 198
380 IFA$="0"THENA$="Z":GOTO430 :rem 185
390 IFA$="{F2}"THENGOSUB2000:RUN:rem 58
400 GOTO900 :rem 102
410 Y=0:G%=1:FORY=0TO152STEP8:FORX=1TO168
STEP8:IFX=1THENX=0:GOSUB900 :rem 120
420 GOSUB900:NEXTX,Y:G%=0:X=88:Y=80:GOTO2
00 :rem 148
430 G%=1:X%=X:FORY=YTOY+Q%:FORX=XTOX+Q%:G
OSUB900:NEXTX,X=X%:NEXTY:G%=0:GOTO200
:rem 110
```

```
500 FORI=1TO50:NEXTI:RETURN :rem 169
550 PRINT"{CLR}{BLK}":POKE36867,174:POKE3
6869,240:POKE36876,0:RETURN :rem 245
560 INPUT"{BLK}NAME:";N$:ON-(N$="")GOTO99
9:PRINT:INPUT"1-DISK 2-TAPE";S:IFS=0G
OTO999 :rem 21
570 POKE157,128:IFS=1THENOPEN15,8,15,"10"
:N$="@":N$:POKE678,8:RETURN:rem 47
580 INPUT"TAPE OFF";O:POKE678,1:RETURN
:rem 133
800 K%=PEEK(W%):S%=K%ORP%(C%):POKEW%,S%:R
ETURN :rem 8
805 POKEW%,PEEK(W%)AND(255-P%(C%)):IFK%<>
PEEK(W%)THENIFA$<>"Z"THENPOKEW%,S%
:rem 223
820 RETURN :rem 122
900 W%=(YAND15)+4096+16*(22*INT(Y/16)+INT
(X/8)):IFW%>7608GOTO200 :rem 66
910 C%=XAND7:GOSUB800:GOSUB805:POKE36876,
250 :rem 132
920 IFA$="Z"THENPOKE36876,160:GOSUB805:ON
-(G%=0)GOTO200:RETURN :rem 3
930 IFABS(EW%)=1THENA$="Z":POKE36876,220:
GOSUB805:GOTO200 :rem 131
950 IFA$="X"ORABS(CW%)=1THENPOKE36876,225
:GOSUB800:IFG%=1THENRETURN :rem 149
960 GOTO200 :rem 106
970 FORI=675TO675+46:READA:S=S+A:POKEI,A:
NEXT:IFS<>5648GOTO1500 :rem 173
980 S=0:RESTORE:FORI=1TO47:READA:NEXT:FOR
I=319TOI+33:READA:S=S+A:POKEI,A:NEXT
:rem 46
990 IFS<>4875 GOTO1500 :rem 213
994 RETURN :rem 134
996 INPUT#15,ER%,ER$,TK%,SC%:PRINT:PRINT:
PRINTER%;ER$,TK%;SC%:IFER%=0THENRETUR
N :rem 85
999 CLOSE 15:END :rem 152
1000 DATA169,3,162,8,160,3,32,186,255,165
,251,166,51,164,52,32,189,255,96
:rem 76
1010 DATA169,0,133,1,169,16,133,2,162,195
,160,29,169,1,32,216,255,96:rem 68
1020 DATA169,0,162,195,160,29,32,213,255,
96 :rem 121
1030 DATA169,0,133,252,169,16,133,253,162
,13,160,0,169,0,145,252,136,208,249
:rem 200
1040 DATA230,253,202,48,6,208,242,160,195
,208,238,141,0,29,96 :rem 235
1500 PRINT"{CLR}ERRORE NELLE DATA'S:STOP
:rem 86
2000 HIGH=20:WIDE=22:ROM=2E15:ADR=36869:O
PEN4,4:PRINT#4,CHR$(8) :rem 18
2010 SIZE=((PEEK(36867)AND1)+1)*8:SC=PEEK
(648)*256:UC=PEEK(ADR)/8AND1
:rem 214
2030 CH=(1-UC)*ROM+(PEEK(ADR)AND7)*1024:L
IM=INT((5+WIDE*8)/6)*6 :rem 70
2040 FORL=0TOHIGH*SIZE-1STEP7:FORC=0TOLIM
:A=0:X%=C/8:BIT=2E(7-C+X%*8):FORR=LT
OR+6 :rem 117
2050 Y%=R/SIZE:CC=PEEK(SC+X%+Y%*WIDE):MEM
=CH :rem 204
2060 IFR<HIGH*8THENIFPEEK(CC*SIZE+MEM+R-Y
%*SIZE)ANDBITTHENA=A+2E(R-L)
:rem 148
2065 GETA$:IFA$="{F4}"ORL>154THENPRINT#4,
CHR$(15):CLOSE4:RETURN :rem 5
2070 NEXTR:PRINT#4,CHR$(128-(C<WIDE*8)*A)
;NEXTC:PRINT#4:NEXTL:CLOSE4:RETURN
:rem 92
```

La biblioteca per ragazzi firmata Jackson.

David Newman - Jacqui Tyler

BATTAGLIE CON IL COMPUTER

A gioco della guerra ha contagiato anche i computer. Qui vengono presentati vari giochi scritti in BASIC per "fare la guerra" sul proprio computer, ma deve essere uno dei seguenti: BBC, Apple, Vic 20, TRS-80, ZX Spectrum o ZX 81.

Cod. 0110 Pag. 48 Lire 9.000

Ian Graham

GIOCHI CON IL COMPUTER

Il libro, parla di giochi con il computer, una volta tanto visti dalla parte del computer, e non dell'utente. In particolare spiega, in modo semplice ma preciso, come fa il computer a giocare, come fa a produrre immagini e suoni durante il gioco e come fa (in genere!) a vincere.

Cod. 006D Pag. 48 Lire 9.000

Brian Reffin Smith

IMPARIAMO A PROGRAMMARE

Il libro fornisce le conoscenze essenziali per incominciare a programmare in BASIC su di un personal computer.

Cod. 018D Pag. 48 Lire 9.000

PRIMI PASSI IN BASIC

Il libro propone l'apprendimento del linguaggio BASIC tramite lo studio e l'esame dettagliato di programmi già scritti.

Cod. 007D Pag. 48 Lire 9.000

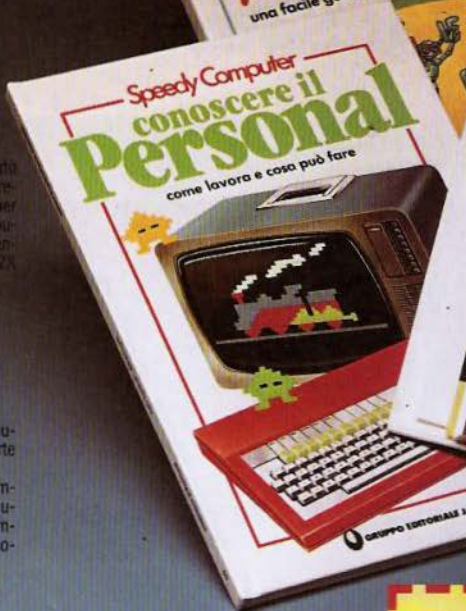
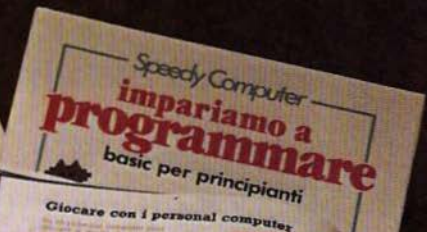
Tony Potter - Ivor Guild

I ROBOT

Dopo una rapida panoramica su che cosa può fare un robot, il libro presenta una serie di robot con funzioni particolari: i robot a braccio, i robot operai, i robot nello spazio, etc.

Affronta poi il problema di come un robot può essere programmato e controllato da un computer, e di come praticamente si realizza un robot.

Cod. 003D Pag. 48 Lire 9.000



Judy Tatchell - Bill Bennett

CONOSCERE IL PERSONAL

Il libro spiega che cosa si può fare con un personal computer, come si usa e come funziona.

Cod. 008D Pag. 48 Lire 9.000

Lynn Miring - Ian Graham

RIVOLUZIONE INFORMATICA

Il volume è rivolto in particolare ai bambini, ma anche a tutti coloro che, presto o tardi, verranno coinvolti dalle nuove tecniche informatiche più come utenti che come operatori.

Cod. 004D Pag. 48 Lire 9.000



GRUPPO
EDITORIALE
JACKSON

ritagliare (o fotocopiare) e spedire in busta chiusa a:
GRUPPO EDITORIALE JACKSON - Divisione Libri - Via Rosellini, 12 - 20124 Milano

CEDOLA DI COMMISSIONE LIBRARIA

VOGLIATE SPEDIRMI

n° copie	codice	Prezzo unitario	Prezzo totale

Totale

Pagherò contrassegno al postino il prezzo indicato più L. 3.000 per contributo fisso spese di spedizione.

Condizioni di pagamento con esenzione del contributo spese di spedizione:

Allego assegno della Banca

Allego fotocopia del versamento su c/c n. 11666203 a voi intestato

Allego fotocopia di versamento su vaglia postale a voi intestato

n° _____

Nome _____

Cognome _____

Via _____

Cap. _____ Città _____ Prov. _____

Data _____ Firma _____

Spazio riservato alle Aziende. Si richiede l'emissione di fattura

Partita I.V.A. _____

ORDINE
MINIMO
L. 50.000

VIC sound

di **A. Kapauan**
trad. e adatt.
di **F. Sarcina**

Questo programma traduce le lettere in musica. Tutto ciò che dovete usare per ottenere i suoni desiderati è una semplice istruzione PRINT.

Print Sound è un programma di utilità scritto in linguaggio macchina, che consente di ottenere melodie musicali da un VIC 20 con semplici istruzioni PRINT. L'uso del programma non richiede alcuna conoscenza del linguaggio macchina.

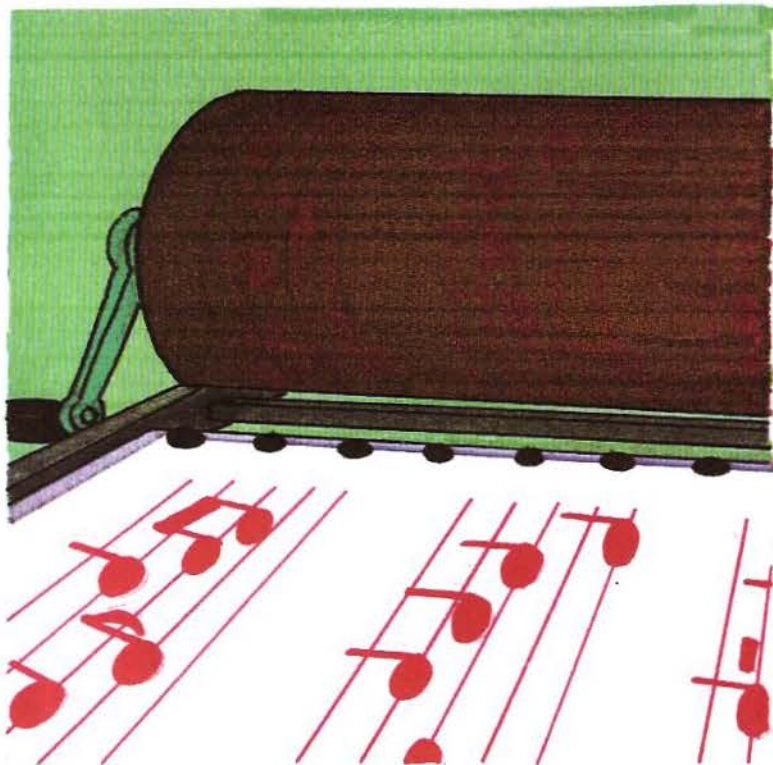
Digitate il listato BASIC che pubblichiamo, prestando particolare attenzione alle istruzioni DATA, e salvate il programma prima di farlo girare. Battendo RUN viene stampata un'istituzione e, dopo una breve pausa, il programma suona alcune note. A questo punto compare il messaggio READY (viene restituito il controllo all'interprete BASIC) e contemporaneamente il computer esegue una breve melodia classica. Se il programma non dovesse funzionare, spegnete il VIC, riaccendetelo, caricate la copia che avevate salvato, listatela e controllate soprattutto i numeri nelle istruzioni DATA. Fate le correzioni necessarie e risalvate il programma.

Se tutto funziona bene, la routine in linguaggio macchina ridefinisce la periferica n. 2 del VIC 20 (che normalmente è la porta RS-232), in modo che ogni stringa di lettere inviata a tale periferica tramite un'istruzione PRINT venga trasformata in una sequenza di note.

Da lettere a note

Se esaminate il listato, noterete che il file logico 1, che nella linea 40 viene aperto per inviare dati alla periferica 2, non viene chiuso. Dopo aver fatto girare il programma digitate il comando PRINT #1, "ABCDEFGHIJKLM" in modo diretto. Udirete in tal modo una scala cromatica, che è la sequenza delle 12 note che si odono suonando, sulla tastiera di un pianoforte, tutti i tasti (bianchi e neri) che compongono un'ot-

Print Sound per VIC 20



tava (per esempio DO, DO diesis, RE e così via fino al DO successivo). La stampa di una lettera da A a Y genera una nota, mentre la Z corrisponde al silenzio e quindi produce una pausa; tutti gli altri caratteri vengono ignorati. Divertitevi a fare tutte le prove che volete, usando il comando PRINT #1, "stringa".

Se volete incorporare in un vostro programma BASIC questa routine in linguaggio macchina, dovette semplicemente includere le linee 500-640, inserendo all'inizio del programma un'istruzione GOSUB630.

L'esecuzione dei comandi PRINT #1 viene completata in genere prima che il VIC finisca di suonare tutte le note. Ciò accade perché le lettere vengono memorizzate in un buffer di 255 caratteri, normalmente riservato al canale di trasmissione della RS-232, mentre una routine che agisce durante l'interrupt (cioè 60 volte al secondo) controlla il trasferimento dei dati ai registri sonori del VIC. Per sincronizzare le note col vostro programma potete esaminare il contenuto della locazione 983, che corrisponde al numero di note attualmente nel buffer. La linea 50 del listato che

pubblichiamo illustra il modo per ottenere la sincronizzazione. Se nel vostro programma desiderate chiudere il file relativo alla periferica 2, fatelo solo quando il buffer è vuoto; comunque nella maggior parte dei casi potete tranquillamente lasciare il file aperto.

La lunghezza delle note che vengono eseguite può essere cambiata modificando il tredicesimo (e terz'ultimo) numero nella linea 570, che normalmente è 10. Tale numero rappresenta la dura-

ta di ogni nota espressa in sessantesimi di secondo. Provate a sostituire 10 con 5 e vedrete con quale velocità il VIC suonerà le vostre musiche!

Dato che la routine in linguaggio macchina risiede nel buffer di cassetta, l'esecuzione di un comando NEW non la cancella. Essa continuerà a funzionare finché non premerete RUN/STOP e RESTORE; in tal caso potrete riattivarla usando il comando SYS 828 e riaprendo (se l'avevate chiuso) il file sulla

periferica 2. Un avvertimento per chi usa il registratore: non dovete eseguire LOAD, SAVE o VERIFY su cassetta mentre la routine è in funzione, perciò prima di effettuare operazioni su nastro premete sempre RUN/STOP e RESTORE per disattivarla. Inoltre tenete presente che tali operazioni, usando il buffer di cassetta, cancellano la routine. Coloro che usano il disk drive non hanno invece questi problemi.

Print Sound per VIC 20

```
10 PRINT"{CLR}{ 4 SPAZI}{RVS} PRINT
   { 2 SPAZI}SOUND " :rem 237
20 GOSUB500 :rem 119
30 PRINT"{GIU'} ORA IL SUONO E' LA"
   :rem 73
35 PRINT" PERIFERICA N.2" :rem 193
40 CLOSE! :OPEN!,2:PRINT#1,"AEHMZLMZZZZZ"
   :rem 187
50 IFPEEK(983)>0THEN50 :rem 16
60 PRINT"{GIU'}-MELODIA DI J.S. BACH-"
   :rem 49
70 PRINT#1,"FHJMKKOMMRQRMJFHJKMOMKJHJFEFH"
   " :rem 52
80 PRINT#1,"AEHKJHJFHJMKKOMMRQRMJFHJCMKJH"
   "FAFEF" :rem 123
90 END :rem 64
500 DATA76,194,3,76,132,3,141,219,3,152,7
   2,165,154,201,2 :rem 236
510 DATA208,47,173,219,3,201,65,48,33,201,
   91,16,29,56,233 :rem 41
520 DATA65,168,185,220,3,172,215,3,192,25
   5,240,249,172,217,3 :rem 193
```

```
530 DATA145,249,200,152,41,255,141,217,3,
   238,215,3,104,168,173 :rem 24
540 DATA219,3,24,96,104,168,173,219,3,76,
   122,242,72,152,72 :rem 97
550 DATA173,218,3,240,15,206,218,3,240,6,
   104,168,104,76,191 :rem 132
560 DATA234,169,0,240,27,173,215,3,240,24
   1,172,216,3,177,249 :rem 188
570 DATA72,200,152,41,255,141,216,3,206,2
   15,3,169,10,141,218 :rem 169
580 DATA3,104,141,10,144,141,11,144,141,1
   2,144,24,144,207,120 :rem 197
590 DATA169,3,141,21,3,141,39,3,169,63,14
   1,20,3,169,66 :rem 148
600 DATA141,38,3,88,96,0,0,0,0,0,183,187,
   191,195,199 :rem 52
610 DATA201,203,207,209,212,215,217,219,2
   21,223,225,227,228,229 :rem 69
620 DATA231,232,233,235,236,237,0
   :rem 128
630 FORI=828TO1013:READJ:POKEI,J:NEXT
   :rem 81
640 POKE36878,15:SYS828:RETURN :rem 96
```

E' IN EDICOLA

Bit,
la prima rivista europea
di personal computer,
software, accessori,
la più prestigiosa
e più diffusa in Italia

con tutta la competenza del



**GRUPPO
EDITORIALE
JACKSON**



7 NOTE BIT: OGGI LA M



AUT. MIN. CONC.

È IN EDICOLA DAL 27 MARZO
LA TASTIERA COMPRESA
NEL PRIMO FASCICOLO



Il 1985 è l'anno mondiale della musica... preparati con **7 note Bit**, la nuova, fantastica opera che in soli 15 fascicoli quattordicinali (ciascuno corredato da una cassetta software per Commodore 64) ti insegna veramente la musica.

Con **7 note Bit**, porti in casa il tuo maestro personale di musica che ti introduce alla teoria e al lessico musicali, ti insegna e ti fa fare pratica sulla tastiera, seguendoti passo passo nelle lezioni ... senza mai annoiarti.

E puoi suonare subito, con l'aiuto del software "interattivo" della cassetta e della speciale tastiera musicale per il tuo 64.

7 note Bit trasforma il Commodore 64 in uno strumento musicale sofisticato ed entusiasmante.

MUSICA SI IMPARA COSÌ.



Con 7 note Bit
vinci 30 straordinari
Commodore Plus/4



Il video del computer ti presenta tutto ciò che ti serve: un metronomo elettronico, il rigo musicale, la tastiera visualizzata con le note corrispondenti ai tasti ... e potrai partecipare "dal vivo" alle sessioni dei musicisti più famosi, suonando insieme a loro.

Prenota in edicola **7 note Bit**: una prestigiosa iniziativa editoriale nata dall'unione della esperienza informatico-divulgativa del Gruppo Editoriale Jackson con la competenza tecnico-musicale **SIEL**.

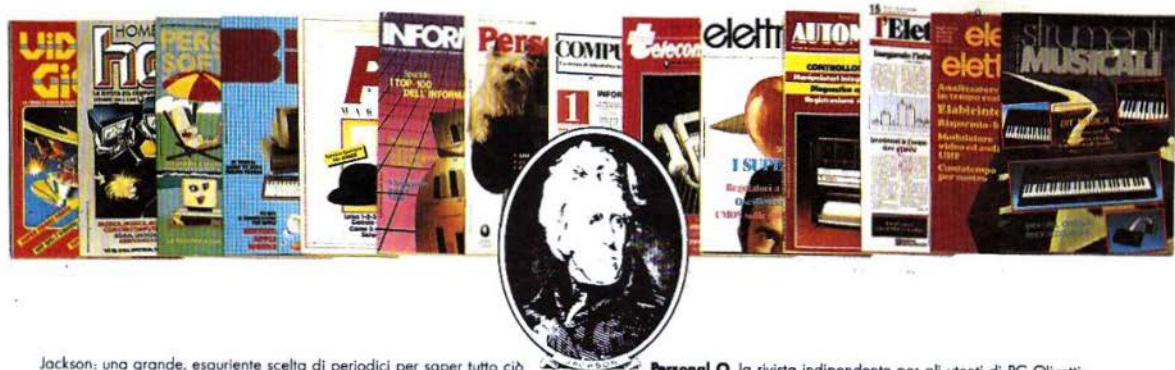
7 note Bit è il "programma musicale" più interessante che c'è: non perdere il primo numero.

Pensa, compresa nel prezzo c'è anche la stupenda tastiera professionale per il tuo Commodore 64.



GRUPPO EDITORIALE JACKSON

Il Jacksoniano sceglie tra 14 top...



Jackson: una grande, esauriente scelta di periodici per saper tutto ciò che è indispensabile su elettronica, informatica e personal computer.

Videogiochi, la guida indiscussa al fantastico mondo dei videogames;

Home Computer, la rivista del computer in casa;

Personal Software, la rivista dedicata al software dei personal computer;

Bit, la prima rivista europea di personal computer, software, accessori, la più prestigiosa e più diffusa in Italia;

Informatica Oggi, il punto di riferimento obbligato per chi si occupa di sistemi EDP e di Office Automation;

PC Magazine, la prima rivista italiana dei sistemi MS-DOS, Personal Computer IBM e compatibili;

Personal O, la rivista indipendente per gli utenti di PC Olivetti;

Compuscuola, la rivista di informatica nella didattica, per la scuola italiana;

Telecomunicazioni Oggi, la rivista di telecomunicazioni e telematica;

Automazione Oggi, il mensile della nuova automazione industriale;

Elettronica Oggi, la più autorevole rivista di elettronica professionale, strumentazione e componenti;

L'Elettronica, il quindicinale di politica industriale, componentistica, informatica e telecomunicazioni;

Elektor, la più diffusa rivista europea di applicazioni e progettazione elettronica.

Strumenti musicali, il periodico di strumenti musicali e computer-music.

Per sottoscrivere abbonamenti compilare il tagliando e inviare in busta chiusa a:

Gruppo Editoriale Jackson S.r.l. - Via Rosellini, 12 - 20124 Milano, allegando assegno o fotocopia della ricevuta di versamento con vaglia postale intestato a GRUPPO EDITORIALE JACKSON - MILANO.

Sì, desidero sottoscrivere un abbonamento a:

- Videogiochi (11 n.) L. 30.000 anziché L. 38.500
 Home Computer (11 n.) L. 31.500 anziché L. 38.500
 Personal Software (11 n.) L. 34.000 anziché L. 44.000
 Bit (11 n.) L. 43.000 anziché L. 55.000
 Informatica Oggi (11 n.) L. 31.000 anziché L. 38.500
 PC Magazine (10 n.) L. 40.000 anziché L. 50.000
 Personal O (10 n.) L. 35.000 anziché L. 40.000

- Compuscuola (9 n.) L. 15.000 anziché L. 18.000
 Telecomunicazioni Oggi (10 n.) L. 28.000 anziché L. 35.000
 Automazione Oggi (11 n.) L. 30.500 anziché L. 38.500
 Elettronica Oggi (11 n.) L. 36.000 anziché L. 44.000
 L'Elettronica (22 n.) L. 44.000 anziché L. 55.000
 Elektor (12 n.) L. 29.000 anziché L. 36.000
 Strumenti Musicali (10 n.) L. 24.000 anziché L. 30.000
 Attenzione per abbonamento all'estero le tariffe devono essere aumentate del 50%

E c'è un super-risparmio a chi si abbona a due o più riviste.

Tutti coloro che sottoscrivono l'abbonamento a due o più riviste godono di un prezzo ulteriormente agevolato, come appare nella seguente tabellina.

Esempio: Bit+Informatica Oggi L. 43.000+31.000=74.000 meno L. 2.000=L. 72.000

Abbonamento

- a 2 riviste L. 2.000 in meno sulla somma dei 2 prezzi d'abbonamento
 a 3 riviste L. 4.000 in meno sulla somma dei 3 prezzi d'abbonamento
 a 4 riviste L. 7.000 in meno sulla somma dei 4 prezzi d'abbonamento
 a 5 riviste L. 10.000 in meno sulla somma dei 5 prezzi d'abbonamento
 a 6 riviste L. 13.000 in meno sulla somma dei 6 prezzi d'abbonamento
 a 7 riviste L. 16.000 in meno sulla somma dei 7 prezzi d'abbonamento

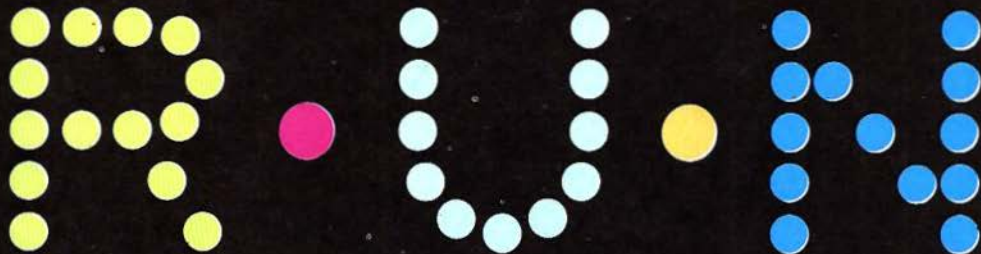
- a 8 riviste L. 20.000 in meno sulla somma degli 8 prezzi d'abbonamento
 a 9 riviste L. 25.000 in meno sulla somma dei 9 prezzi d'abbonamento
 a 10 riviste L. 30.000 in meno sulla somma dei 10 prezzi d'abbonamento
 a 11 riviste L. 35.000 in meno sulla somma degli 11 prezzi d'abbonamento
 a 12 riviste L. 40.000 in meno sulla somma dei 12 prezzi d'abbonamento
 a 13 riviste L. 44.500 in meno sulla somma dei 13 prezzi d'abbonamento
 a 14 riviste L. 50.000 in meno sulla somma dei 14 prezzi d'abbonamento

- Allego assegno n° _____ della Banca _____
 Ho effettuato versamento sul c/c postale n. 11666203 intestato a Jackson - Milano di cui allego fotocopia della ricevuta
 Vaglia postale e allego fotocopia ricevuta

Nome _____
 Cognome _____
 Azienda _____
 C.A.P. _____ Città _____
 Via _____



**GRUPPO
EDITORIALE
JACKSON**
San Francisco-Londra-Milano



Recensioni Software

di **M. Cristuib Grizzi**

Tower of Evil per Vic 20

Thorn Emi sftw.



Un gioco per VIC 20, con espansione di memoria da 8 Kbyte, che comprende le caratteristiche di un adventure, frammiste a quelle di un buon "arcade".

Molti anni fa il nostro eroe Andros fu scacciato dalla corte del re Salimos e destinato a non fare più ritorno fino a che avesse ritrovato il tesoro perduto del re e la bella principessa Diana rapita da un mago cattivo.

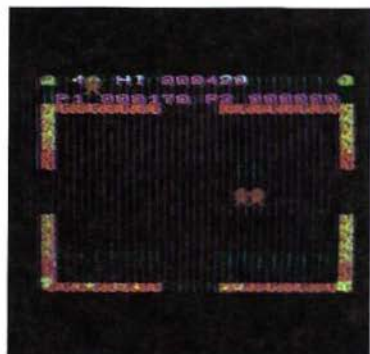
Andros dispone di un potente aiuto da parte di un vecchio amico, il mago di corte, che ha il potere di lanciare palle di fuoco dalle sue dita.

Il nostro eroe ha viaggiato fino alla Tower of Evil (Torre del Male), abitazione del mago cattivo. Dovrà quindi esplorarne le varie (e numerosissime...) stanze, uccidendo man mano i mostri incontrati. Andros dovrà ritrovare i vari preziosi costituenti il tesoro e recuperare la chiave che gli permetterà di passare al piano successivo. Al settimo piano Andros potrà finalmente depositare il tesoro e salvare la bella principessa imprigionata.

La grafica è molto buona, con "mostri" diversi per ogni piano della Tower of Evil, e la velocità di gioco è quella tipicamente ottenibile solo con l'utilizzo del linguaggio macchina.

Il gioco è utilizzabile sia da tastiera che attraverso il joystick ed ha l'opzione di scelta tra uno o due giocatori.

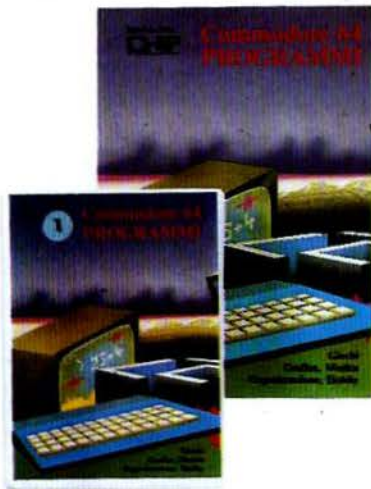
Racchiudendo in sé le migliori caratteristiche dei giochi di avventura, unitamente alla velocità e grafica in alta risoluzione degli "arcade", Tower of Evil è un gioco ideale per passare ore divertenti in compagnia del proprio VIC 20.



**Questo programma
può essere ordinato
utilizzando
il coupon di pag. 82**

di **M. Cristaib Grizzi**

Commodore 64: programmi ediz. Tecniche Nuove



Una raccolta di programmi per C64 che dedica il più ampio spazio possibile ai listati, facendoli precedere da un sintetico commento sul funzionamento del programma e dalla sua breve descrizione.

Si spazia dai giochi che richiedono rapidi riflessi e prontezza con il joystick, a quelli di carattere più meditativo, che richiedono doti logiche e di ragionamento per essere risolti con successo. Troviamo infatti i classici "Lancio con il Paracadute", "Bombardiere", "3D Maze" ed i giochi di ragionamento come "Reversi", "Torre di Hanoi", "Rotazione di numeri"...

Una seconda parte viene dedicata alla grafica, con un programma che visualizza sul video la carta geografica dell'intero globo, alla musica, con un programma che trasforma il C64 in un completo sintetizzatore musicale, ed all'utilizzo casalingo, con un programma generato-

re di calendari ed un'agenda telefonica.

L'ultima parte del libro è dedicata alla "scienza ed hobby", e comprende vari programmi di utilità, quali "Bioritmo", "Elettronica" e "Distanze".

I listati sono stati ottenuti con stampante a matrice di punti senza alcuna codifica, per cui il listato raffigura esattamente quanto deve apparire sul video listando un programma.

Al volume è allegata una cassetta contenente, già registrati, tutti i programmi presentati nel libro, al fine di evitare il noioso lavoro di battitura degli stessi.

Commodore 64: guida all'uso di Ian Sinclair ediz. Tecniche Nuove



Molte persone acquistano il C64 e si trovano subito davanti a notevoli difficoltà nell'apprenderne l'uso, a causa della scarsità e superficialità delle infor-

mazioni riportate nel manuale allegato alla confezione della macchina.

Questo libro si propone di supplire alle deficienze del manuale allegato al C64, fornendo tutte le informazioni necessarie per poterne iniziare la programmazione. In particolare il primo capitolo si occupa delle caratteristiche fondamentali della macchina, quali la tastiera, i collegamenti con le unità periferiche ed il procedimento per memorizzare e caricare programmi da nastro o disco.

Il secondo capitolo ed i seguenti sono invece dedicati al BASIC, e guidano per mano l'utente attraverso tutte le istruzioni utilizzabili, descrivendo in grande dettaglio la funzione ed il risultato provocato da ogni istruzione.

Assunto che a questo punto della lettura si sia raggiunta una accettabile confidenza con il BASIC, il libro passa ad esaminare la grafica con caratteri programmabili ed utilizzo di Sprite. Una ulteriore parte del libro è dedicata al suono e mostra come poter ottenere effetti sonori attraverso un'accorta programmazione.

L'ultima parte tratta in particolare di come accedere al codice macchina attraverso il BASIC e di tutti i tipi di periferica collegabili al C64, fornendone una sommaria descrizione.

L'autore, Ian Sinclair, ha scritto già molti libri sul C64, rinomati per chiarezza d'esposizione e ricchezza di contenuto.

Il meglio dal Commodore 64 di D. Lawrence ediz. Mondadori

"Il meglio dal Commodore 64" è dedicato a chi voglia ottenere dal proprio computer non solo un emulatore di console per videogiochi, ma anche una serie di applicazioni veramente utili, come quelle casalinghe.

Vengono infatti presentati diciannove programmi utilizzabili per applicazioni

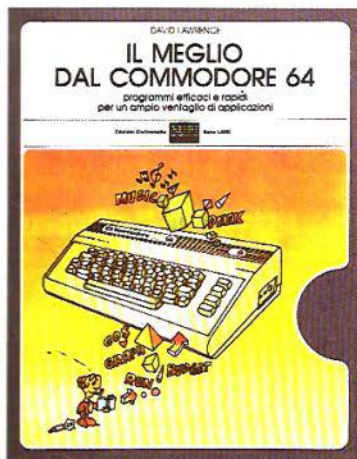
E' IN EDICOLA

"intelligenti" del computer: tra i più interessanti troviamo un archivio elettronico, un programma di gestione del bilancio familiare, un semplice word processor ed un programma per disegnare.

Tutti i programmi sfruttano la tecnica di programmazione modulare, per cui ognuno di essi è sfruttabile in sé, oppure è inseribile, insieme ad altre routine, in un unico e lungo programma che esegue tutte le applicazioni proposte.

Ciascun modulo è ampiamente e dettagliatamente commentato, rendendo estremamente facile comprendere la struttura e le funzioni delle varie linee del programma.

Un buon aiuto, quindi, per chi desidera imparare la programmazione del C64, sfruttandone allo stesso tempo le migliori caratteristiche.



I libri italiani vengono offerti con il 10% di sconto sul prezzo di copertina.

**i libri recensiti
possono essere
ordinati utilizzando
il coupon a pag. 82**

SOFTWARE

CON TUTTA
LA COMPETENZA
DEL



**GRUPPO
EDITORIALE
JACKSON**

La prima
rivista europea
di software
per personal computer

input-output

Piccoli annunci

Scambio, vendo software per Commodore 64 su disco. Silvano Bompiers - Via Baccaglioni, 8 - 46040 Monzambano - Tel. 0376/845372

Cerco stampante MPS801 in buono stato, solo se occasione. Telefonare dopo le ore 20.30 a: Salvatore Impalato - Via Consalvo, 148 - 80125 Napoli - Tel. 081/627459 627912

Vendo cassetta per C64 con Turbotape (per caricare più velocemente i programmi) e stupendi videogames (Puc Man, Frogger, Calcio, Gobbo, ecc.) a prezzo variabile da L. 1.000 a L. 5.000 l'uno a seconda del tipo di gioco, oppure 10 giochi a L. 30.000. Adrian Massimo - Via Aquileia, 63 - 34070 Villesse (GO) - Tel. 0481/91095

Vendo VIC 20 completo di manuale, trasformatore e imballo più 2 cartidge (Cosmic Cruncher e Visible Solar System) come nuovi. Prezzi modici. Telefonatemi ore pasti. Giuseppe Mazzacane - Via G. Fanelli, 226/40 - 70100 Bari - Tel. 481614

Causa passaggio a sistema superiore vendo in blocco oltre 150 programmi per Commodore 64 con molte utility (Superbase, Easy Script, The Last One...) Francesco D'Andrea - Via G. Messina, 33 - 74100 Taranto - Tel. 099/314517

È nato il Commodore Club della bassa bergamasca, con scopo di scambio di listati e programmi. Scrivere a: Commodore Club - Casella Postale - 24058 Romano Lombardo (BG)

Vendo o cambio giochi per C64, tra i quali originali da sala giochi: Donkey Kong, Zaxxon, Popeye, Il Gobbo, Pole Position, Decathlon, Baseball, Congo Bongo, Buck Rogers, Pitfall I, Frogger, QBert, Scramble, Pojan e tanti altri. Oreste Antignano - Via Luna e Sole, 23 - 07100 Sassari - Tel. 079/291417

Compro C64 distrutto, per pezzi di ricambio. Vendo libro "Alla scoperta del VIC 20" imballato, disco programmi del libro, Sea Wolf, Jupiter Lander, Road Race, interfaccia registratore a L. 10.000 l'uno. Mario Comuzzo - Via Pola, 1 - 33010 Tavagnacco (UD) - Tel. 0432/680044

Solo per zona Novara e dintorni: compro espansione 32 Kbyte o 16 Kbyte RAM a prezzi modici; cerco metodo per duplicare le 3 Kbyte s.e. e 8 Kbyte e le cartidge. Cambio giochi solo se di alto livello. Tutto per VIC 20. Luigi Sempio - Via Podgora, 11 - 28100 Novara - Tel. 0321/31327

Eccezionale! Vendo VIC 20 più trasformatore, cavo antenna, 2 manuali di istruzione (inglese e italiano), "20 programmi per VIC 20" della Curcio, "Giochi per il vostro VIC 20" della Jackson a L. 150.000. Vendo anche Dragonfire, Gorf, Gardenwars. Franco Peloso Gaspari - Calle Annunziata, 2 - 33053 Latisana (UD) - Tel. 0431/50383

Per C64 compro programmi didattici ed utility, scolastici ed extra, purché siano ad alto livello di risoluzione grafica e capacità di calcolo. Compro inoltre il programma "Scacchi" e "Totocalcio". Inviatemi le vostre liste con i vostri, spero, modici prezzi. Rispondo solo ai migliori offerenti. Fabio Bianchini - Via Monte Solder, 5 - 31044 Montebelluna (TV) - Tel. 0423/23469

Vendo i libri "Alla scoperta del VIC 20" e "Giochi per il vostro VIC 20" e i cartidge "Avenger e The Count". Marco Lovera - Via Del Moncone, 36 - 21010 Cardano al Campo (VA) - Tel. 261618

Vendo programmi per VIC 20: cassetta con 70 programmi in versione base L. 30.000 oppure vendo cassetta con programmi ad 8 Kbyte di espansione, 25 tra giochi ed utility, a L. 30.000. Affrettatevi. Luciano Baglioni - Via Della Verena, 20 - 00141 Roma - Tel. 06/8185710

Vendo VIC 20 più 3 cartucce a L. 210.000. Massima serietà. Vendo anche cassette con 168 programmi a L. 170.000. Prima di comprarlo, venite a provarlo. Luca Lorenzini - Via Lumumba, 11 - 41011 Campogalliano (MO) - Tel. 525861

Cerchi software per il tuo CBM 64? Mandami il tuo indirizzo e ti farò ricevere la mia lista con più di 300 programmi. Bruno Rudella - Via Verdi, 31 - 24040 Acene (BG) - Tel. 035/878594

Vendo CBM 64 più registratore, plotter 1520, monitor Philips (fosfori verdi), scrivania con cassetiera estraibile (con rotelle) laccata bianca, numerosi giochi e utilities, joystick, il tutto a L. 1.300.000 non trattabili. Galizia Antonio - Bari - Tel. 080/415903 209214 (ore ufficio)

Vendo VIC 20 più registratore C2N, joystick, 150 programmi, 3 manuali e 2 libri pieni di listati, 10 riviste, cassetta pulisci testine; il tutto in buono stato al prezzo di L. 390.000. Telefonare ore pasti o dalle 20 in poi. Antonio Presepri - Via Sassari, 4 - 47100 Cesenatico (FO) - Tel. 83967

Vendo Videopac G7000 più cartuccia con manuale programma Assembler, 2 cartucce (Caccia al Tesoro e Invaders), il tutto a L. 300.000. Luca Costoro - Via G.B. Vico, 12 - 05100 Terni - Tel. 0744/454284

Vendo, cambio oltre 1.000 programmi per C64. Tutte le novità, prezzi modici. Telefonare o scrivere a: Mario Berardo - Via Vittorio Emanuele, 265 - 12042 Bra (CN) - Tel. 0172/421703

Possesso molti giochi per il C64 e vorrei scambiarli con altri giochi. Mandatemi la vostra lista ed io vi spedirò la mia. Risponderò a tutti. Tristano Magnani - Via Galilei, 2 - 01029 Orte Scalo (VT) - Tel. 490327

Vendo, scambio software per CBM 64. Possesso circa 600 programmi di giochi, utility e gestionali. Prezzi stracciati. Telefonare ore pasti. Olderio Caviglia - Via D. Carbone, 4 - 15050 Villalvernia (AL) - Tel. 0131/83150

Cambio registratore Hi-Fi nuovo Yamaha K200B con Commodore 64. Conguaglio per eventuale differenza. Telefonare dopo le 18. Fernando Casula - Via Franzone, 126 - 25100 Brescia - Tel. 316520

Vendo per VIC 20 "Donkey Kong", "Cosmic Cruncher", "Avanger", "Visible Solar System", "Jupiter Lander" più 6 libri "Alla scoperta del VIC", "Sinfonia per un computer", "Giochi per VIC", "Guida al VIC", "Giocare con il BASIC", "Commodore VIC 20" e 3 libri su ZX81. Carlo Carozzo - Via Agostino Gioia, 60 - 70054 Giovinazzo (BA) - Tel. 080/931969

Programmi per Atari 800 XL. Moltissimi programmi su disco e cassetta. Carlo Volpi - Via Enrico Cravero - 00100 Roma - Tel. 06/5117950

Attenzione! Ad Augusta è nato un club per gli utenti di tutti i computer. Il suo nome è "Home & Personal Computer Club". Potete possedere software completamente autentico con lo sconto del 60%. Non vi sembra fantastico? Scriveteci per informazioni. Michele Antonioy - Via Mazzini, 1 - 96011 Augusta (SR) - Tel. 0931/9929

Vendo per VIC 20 espansione 3-8-16 Kbyte selezionabili, VIC-FORTH più 3 Kbyte RAM, 4 slot, monitor L.M., Reference Guide, Guida al VIC EVM, e molti altri libri e programmi. Alberto Baldan - Via S. Gallo, 168 - 30126 Lido di Venezia - Tel. 041/461080

Vendo programmi per VIC 20 e C64; a chi richiede il catalogo invieremo la tessera e l'opuscolo del nostro Commodore Computer Club. Gian Luca Rivelli - Via Dante Alighieri - 17025 Loano (SV) - Tel. 669773

Vendo VIC 20 più libro (46 programmi per giochi), trasformatore, 4 cartidge ("VIC Avenger", "Rat Race", "Star Post", "Gorf"), 2 cassette di giochi ("Catcha Snatcha" e "Pac Man"), manuale d'istruzioni inglese e italiano. Tutto in ottime condizioni. Vero affare! Marco Paolucci - Via Appiani 20030 Senago (MI) - Tel. 9980143

Cambio software per Commodore 64. Vendo anche giochi a L. 2.000 (un vero affare!). Romano Vallarin - Via G. Pardini, 5 - 20090 Trezzano s/n (MI) - Tel. 02/4454981

Possesso oltre 100 programmi (games ed utility): Voce, Flipper, Decathlon, Star Trek, Pole Position, Manic Miner, Fort Apocalypse, Calcio, Suicidestrike, Crazy Kong, Cuddly Curburbs, Skramble, Aztec Challenge, Pipeline e tantissimi altri programmi, tutti eccezionali. Aspetto richieste per scambi, rispondo sempre. Filippo Morri - Via Olivella, 5 Serravalle - 47031 Repubblica di S. Marino - Tel. 900245

Vendo cartidge per VIC 20 "Visible Solar System" a L. 20.000, spese postali comprese. Scrivere per accordi a: Gianluca Bianchi - Via Grazioli, 17 - 29012 Caorso (PC) - Tel. 821457

Vendo VIC 20, per sbaglio corrispondenza, ancora imballato (con manuale in italiano) più cartuccia gioco "Sea Wolf" con manuale "Giochi per il VIC 20" e 20 listati di giochi a L. 220.000. Telefonare ore pasti. Maurizio Gatti - Via Mario Abbiate, 100 - 13100 Vercelli - Tel. 0161/63494

Vendo VIC 20 nuovissimo con registratore, 14 cassette contenenti 90 giochi, libro "Alla scoperta del VIC 20" e un libro con 30 programmi. Il tutto, con soli due mesi di vita, a L. 230.000. Stefano Fiorentini - Via Servulio Prisco, 1 - 00175 Roma - Tel. 7475811

Vendo, cambio giochi, data base, linguaggi, tabelle elettroniche, materiale per suono e grafica, utility varie, word processor, copiatori, compilatori, didattica per CBM 64 a basso prezzo (vasta gamma). Simone Lozza - Via Risorgimento, 57 - 20098 S. Giuliano Milanese - Tel. 9840050

Vendo, cambio più di 500 programmi in LM per CBM 64: a L. 5.000 i giochi e L. 6.000 le utility. Per informazioni rivolgersi a: Marco D'Amico - Via Pio IX, 125 - 00100 Roma

Vendo microcomputer Sinclair ZX80: valido supporto per chi si rivolge per la prima volta alla programmazione. Prezzo di realizzo. Come nuovo. Pasquale Meli - Piazza S. Agostino, 38/14 - 19100 La Spezia

Vendo VIC 20 più registratore Commodore 1530 C2N, 2 cartidge, cassetta giochi al prezzo unico di L. 340.000. Prezzo trattabile. Annuncio sempre valido. Vendo anche separatamente. Scrivere o telefonare. Luigi Beltramini - Via Della Stradale - 5 Felice Circeo (LT) - Tel. 0773/536241

Cambio software per C64. Rispondo a tutti, annuncio sempre valido. A. Mario Campus - Via Genova, 40 - 07100 Sassari

Vendo o cambio circa 1.500 programmi per C64. Vasto assortimento gestionali, completi di manuali d'istruzione in italiano e inglese e circa 1.300 giochi vari. Angelo Settembrini - Via Cassanese, 194 - 20090 Segrate (MI) - Tel. 02/2136514

input-output

Piccoli annunci

Per Commodore 64 disponendo di innumerevoli programmi vendo a L. 10.000 cassetta con 5 giochi a scelta, anche su disco. Annuncio sempre valido. Scrivere a: Parbuono Ivano - Via A. di Cambio, 4 - 37138 Verona - Tel. 045/568649

Linguaggio macchina per C64. Mi rivolgo a chi non è commerciante del proprio sapere. Non trovando testi, da anni, sul mercato, vorrei comprendere come altri possano invece essere delle teste di ponte. Sono disposto al lavoro o a contatti epistolari. Passo tutto il mio tempo a disassemblare. Mauro Marsilli - Via Giambellino, 44 - 20146 Milano - Tel. 02/4235971

Cerco disperatamente memoria 3, 8, 16 Kbyte per VIC 20 a prezzo accessibile. Scrivere o telefonare a: Nicola Tiso - Via Mascagni, 33 - 36100 Vicenza - Tel. 0444/562467

Vendo e cambio oltre 200 programmi per C64, fra cui molti bellissimi, a L. 4.000 più 2.000 per cassetta. Richiedere catalogo a: Vittorio Lipari - Via Barbiano, 1/3 - 40136 Bologna - Tel. 051/332197

Si è costituito il Club 64 in provincia di Padova per scambi e vendite. Ultime novità. Club 64 - Casella Postale 6 - 35044 Montagnana (PD) - Tel. 0429/82469

Vendo VIC 20 più espansione programmabile 4-8-16 Kbyte, registratore C2N, corso BASIC su cassette, 2 manuali in italiano, molti programmi su cassette a L. 300.000 solo in zona Castelli Romani. Franco Bellardini - Via A. De Gasperi, 140 - 00045 Genzano (RM) - Tel. 06/9399584

Vendo ZX81, espansione 16 Kbyte, tastiera tasti premanti, 2 manuali per BASIC, 1 manuale per Assembly, un centinaio di programmi tra cui: Scacchi, Volo Simulato, Assembly/Disass., Turbo-sasse, Compiler, File, data base e tanti altri a L. 200.000. Solo zona Castelli Romani. Franco Bellardini - Via A. De Gasperi, 140 - 00045 Genzano (RM) - Tel. 06/9399584

Vendo CBM 64, perfettamente funzionante, così come il registratore dedicato C2N; programmi di utilità "Tool 64" (originale, in cartuccia). "Simons' BASIC", un Archivio e un word processor (anch'essi originali con manuale di istruzioni); 2 giochi in cartuccia, 18 cassette di giochi (quasi tutti in LM); un libro per la programmazione, la "Guida di riferimento del programmatore", riviste e molti listati BASIC e LM, 1 joystick. Il tutto a prezzo di concorrenza. Pasquale Sansò - Via G. Gentile, 55/c - 70126 Bari - Tel. 080/491345

Cambio o vendo software per C64. Sono in possesso di molti bei videogiochi ed utility (Zaxxon, Baseball, Basket, word processor, Mod. 740, ecc.). Massima serietà. Corrado Ferrari - Via Europa, 40 - 20097 San Donato Milanese - Tel. 02/5274458

Vendo oltre 100 programmi per CBM 64 su cassetta a L. 7.000 cadauno: Decathlon, Koala Painter, Manic Miner, Pitstop, Forza 4, Poker, Pac Man, Diete, Baseball, Donkey Kong e tanti altri, tutti con Turbo Tape. Annuncio sempre valido. Giuseppe Tarantino - Largo Venezia, 144 - 88074 Crotona (CZ)

Vendo VIC 20 più espansione 3,8-16 Kbyte con registratore (Sanyo) e interfaccia joystick (Spectravideo, con auto-fire), 10 cassette giochi e 2 manuali. Il tutto ancora imballato a sole L. 300.000. Fabrizio Rossi - Via Fabio Filzi - 20010 San Giorgio (MI) - Tel. 0331/400174

Cambio auto radiocomandata Redel 45 Mod. Lancia Beta Montecarlo Turbo con espansione 16 Kbyte. Valore auto L. 60.000, condizioni perfette. Oltre alla macchina offro L. 30.000. Per informazioni scrivere a: Gianluca Mentella - Via Capocci, 3 - 03043 Cassino (FR) - Tel. 0776/24636

Vendo 40 programmi per CBM 64, tutti in LM, registrati con il Turbo Tape (compreso in cassetta) ad ottimi prezzi (ognuno L. 3.000, 6 L. 15.000). Possiedo programmi come Pac Man, Pole Position, ecc. Gianluca Flamini - Via Teano, 1 - 00177 Roma - Tel. 06/290194

Per Commodore 64 vendo centinaia di programmi su nastro o disco: Turbo Disk, Super Base, Easy Script, Clone, Unguard, Dallas, Decathlon, Magazzino, Gyrrus, FORTH, Logo, ecc. Per la lista rivolgersi a: Stefano Dominioni - Via Tommaseo, 18 - 21100 Varese - Tel. 0332/229909

Vendo VIC 20 più registratore C2N, 1 cartridge, 120 programmi tra cui: Pac Man, Donkey Kong, 40 Colonne, Contabilità, Rubrica Telefonica, Totocalcio, Formula 1, ecc. Il tutto a L. 280.000. Jacopo Mangiavacchi - Via S.A. Merici, 70 - 00162 Roma - Tel. 8323095

Vendo programmi per C64 su cassetta da L. 1.000 a L. 2.000, anche ultime novità (Decathlon, Wimbledon, Pitfall 2). Richiedere lista a: Marco Romani - Via G. Valli pal. B1 - 00149 Roma - Tel. 06/5265121

Giochi, giochi, giochi vendo giochi per il vostro VIC 20, tra cui: Frogger, Pac Man, Scacchi, Briscola, Invaders, Enduro, Miliepedi, senza espansione: tutto a L. 20.000. Vendo altri programmi da L. 1.500 a L. 2.500, anche utility. Gianluca Bonfanti - Piazzale Boito, 1 bis - 43100 Parma - Tel. 0521/36452

Cambio Intellevision più 9 cassette e joystick (valore commerciale L. 800.000) con floppy disk drive 1541, oppure con stampante plotter a colori 1520 con aggiunta di qualche cassetta gioco, il tutto per VIC 20. Oppure vendo il tutto, in buone condizioni e in imballo originale, a L. 350.000. Gianluca Flamini - Via Teano, 1 - 00177 Roma - Tel. 06/290194

Vendo programmi in LM per Commodore 64 a meno di L. 5.000. Spedisco i programmi in contrassegno, su disco o cassetta. Raffaele Celi - Piazza A. Garibaldi - 88063 Catanzaro Lido (CZ) - Tel. 31889

Vendo ottimi video games e utility in turbo su cassetta. Prezzi modici, assicurata massima serietà. Richiedere liste a: Nino Riggio - Via Esseneto, 65 - 92700 Agrigento

Cambio e cambio per Commodore 64 giochi e utility, tra cui: Calcio, Corsa, Enduro, You Speak English e tanti altri giochi ed utility. Per la vendita, i prezzi sono bassissimi. Antonio Fiorito - Via Pasubio, 54 - 71017 Torremaggiore (FG) - Tel. 0882/281899

Vendo molti programmi su cassetta (700 circa) per VIC 20 inespanso o espanso 8-16 Kbyte e per Spectrum 16-48 Kbyte. Sono i migliori programmi sul mercato, tutti in LM. Vendo listati per tutti i computer. Pierluca Colfanetti - Via Prealpi, 7 - 22060 Arosio (CO) - Tel. 031/761785

Cambio giochi ed utility per C64. Inviare lista a: Renzo Lenzi - Piazzale Montello, 8 - 57100 Livorno - Tel. 0586/804380

Vendo VIC 20 come nuovo (6 mesi di vita) più registratore C2N, cartridge VIC Avenger, 3 libri per VIC 20, oltre 150 videogiochi (molti in LM) e utilities in cassette. Prezzo incredibile solo L. 250.000. Simone Baglieri - Via Platani, 1 - 93012 Gela (CL) - Tel. 0933/916454 (ore 14.00 - 17.30)

Novità per Totocalcio e Totip su C64: finalmente un modo di giocare che può fare solo un computer in grado di operare tutte le colonne possibili. Ottimizzato su 11 triple; a L. 100.000 su disco o cassetta. Ernesto Chianette - Via Nemorense, 39 D - 00199 Roma - Tel. 06/8433884

Per Commodore 64 vendo oltre 800 programmi tra i migliori a prezzi bassi su nastro o disco. Chiedere lista gratuita. Gabriele Fanelli - Via C. Zaccagnini, 129 - 00128 Roma - Tel. 06/6151345

Vendo VCS 2600 Atari, con tre cassette in regalo, a sole L. 190.000 oppure cambio con espansione 32 Kbyte RAM per VIC 20. Vero affare, scrivete. Stefano Chelli - Via Lorenzo Quartieri - 55042 Forte dei Marmi

Per C64 vendo, cambio ogni tipo di programma su disco o nastro. Offro e chiedo la massima serietà. Scrivere o telefonare a: Maurizio Sternieri - Via Gobetti, 10 - 41012 Carpi (MO)

Per CBM 64 vendo, compro, cambio software su nastro e disco. Vasto elenco programmi, ultime novità. Roberto Restelli - Via Pesa, 34 - 20080 Tavazzano (MI)

Vendo, cambio software di qualsiasi genere (games, utilities) e molti altri a sole L. 3.000. Per ricevere la lista inviare L. 1.000; i programmi sono tutti per VIC 20 e C64. Enrico Cuoco - Via Milano, 125/12 - 20024 Garbagnate (MI) - Tel. 02/9959088

Vendo per VIC 20; cartuccia "The count" a L. 25.000; 1 cassetta con 30 giochi, 4 utilities (Grid Runner, Lamas, Amok, Blitz, Dieta, Calorie, ecc.) a L. 20.000 più spese postali in contrassegno. Per accordi telefonare dalle ore 19.00 alle 20.00 o scrivere a: Giuseppe Lumediluna - Via Marco Zoppo, 4 - 35100 Padova (PD) - Tel. 049/603385

Vendo VIC 20 più registratore, 4 cassette, 10 cartucce, joystick Spectravideo. Tutto quasi nuovo e in imballi originali a L. 300.000. Giò Dal Negro - Via Sansovino, 8 - 37100 Verona - Tel. 045/576375

Cambio, vendo per Commodore 64 bellissimi giochi ed utilities in LM a prezzi stracciati. Annuncio sempre valido. Scrivere o telefonare dopo le ore 20.30. Roberto Zavatta - Via G. Verdi, 15 - 47030 S. Mauro Pascoli (FO) - Tel. 0541/931114

Vendo programmi per VIC 20 e C64. Massimiliano Gulotta - Via Lago di Bracciano - 58022 Follonica (GR) - Tel. 0566/53069

Il Club Commodore Special vende programmi per C64 e VIC 20. Si accettano iscrizioni per L. 3.000, con 2 magnifici programmi in regalo. Si spediscono anche cassette. Stefano Bricchi - Via Trento, 35 - 58022 Follonica (GR) - Tel. 0566/43830

SUPERCOMMODORE

CEDOLA DI ORDINAZIONE - LIBRI
 da compilare e spedire in busta chiusa a
 J.soft - Via Restelli, 5 - 20124 Milano - Tel. 6888228

SOFTWARE

TOWER OF EWIL

cod. DTEMV04

L. 15.000

Ordino i seguenti libri per un importo totale di L.+ L. 2.000
 come contributo fisso per spese di spedizione

Cod.Cod.Cod.
 Cod.Cod.Cod.

- Contanti allegati Assegno allegato n°
- Ho spedito l'importo a mezzo vaglia postale
- Ho versato l'importo sul cc. n° 19445204 intestato a J.soft - Milano
- Pagherò in contrassegno al postino al ricevimento delle cassette

Nome

Cognome

Via

Città C.A.P. Prov.

Se richiesta fattura:
 Cod. F. e P. Iva

Data

Firma

Per i residenti all'estero - pagamento anticipato (vaglia o versamento su ns. ccp)

SUPERCOMMODORE

CEDOLA DI ORDINAZIONE - LIBRI
 da compilare e spedire in busta chiusa a
 J.soft - Viale Restelli, 5 - 20124 Milano - Tel. 6888228

LIBRI

Commodore 64: Programmi cod. ATNC002 L. 40.500

Commodore 64:
 Guida all'uso cod. ATNC008 L. 13.500Il meglio dal
 Commodore 64 cod. AMOC007 L. 16.200

Ordino i seguenti libri per un importo totale di L.+ L. 2.000
 come contributo fisso per spese di spedizione

Cod.Cod.Cod.
 Cod.Cod.Cod.

- Contanti allegati Assegno allegato n°
- Ho spedito l'importo a mezzo vaglia postale
- Ho versato l'importo sul cc. n° 19445204 intestato a J.soft - Milano
- Pagherò in contrassegno al postino al ricevimento dei volumi (valido solo per i soci in Italia)

Nome

Cognome

Via

Città C.A.P. Prov.

Se richiesta fattura:
 Cod. F. e P. Iva

Data

Firma

Per i residenti all'estero - pagamento anticipato (vaglia o versamento su ns. ccp)

SUPERCOMMODORE è bello, però... (ovvero suggerimenti, idee, critiche, richieste e tutto ciò che vi passa per la testa).

.....

.....

.....

.....

.....

Nome Cognome

Via C.A.P.

Città Tel.

SUPERCOMMODORE

La rubrica INPUT/OUTPUT è gratuita ed aperta a tutti i lettori. Chi desidera comprare, vendere o cambiare hardware o software può inviare il tagliando a J.soft - Viale Restelli, 5 - 20124 MILANO

INPUT/OUTPUT

- COMPRO VENDO CAMBIO
- VIC 20 C 16 C64 PERIF. SOFTWARE

.....

.....

.....

Nome Cognome

Via C.A.P.

Città Tel.



AFFIDA I TUOI DATI A UN SUPPORTO SICURO

Come editori di software, abbiamo sentito l'esigenza di utilizzare, per la produzione dei nostri programmi, un supporto particolarmente affidabile. Dopo severi ed accurati test abbiamo operato la scelta. Siamo lieti di proporlo con il nostro marchio a chiunque desideri lavorare con la nostra stessa tranquillità. Floppy disk da 5" 1/4, singola faccia, doppia densità, in confezione da 10 dischetti. Ordine minimo 10 dischetti. Ordini superiori solo multipli di 10 secondo la seguente scala di prezzi:

- 10 dischetti	L. 5.000 cad.
- da 20 a 50 dischetti	L. 4.700 cad.
- da 60 a 100 dischetti	L. 4.400 cad.
- da 110 dischetti e oltre	L. 3.900 cad.

I prezzi sono comprensivi di IVA e spese di spedizione.

Per ordinare ritagliate e spedite il tagliando sotto riportato a
J.soft - Viale Restelli, 5 - 20124 Milano
Tel. 02/6888228 - 683797 - 6880841 - 6880842 - 6880843

10 J disk


J.soft

Via Roselli, 12 - 20124 - Milano - Tel. 02/6888228-683797-6880841-6880842-6880843

CEDOLA DI ORDINAZIONE OFFERTA DISCHETTI

Da compilare e spedire in busta chiusa a
J.soft - Viale Restelli, 5 - 20124 Milano
Tel. 02/6888228 - 683797 - 6880841 - 6880842 - 6880843

Ordino i seguenti dischetti, in confezione da 10 pezzi cad., per un importo totale di L. IVA e spese di spedizione incluse.

- N. dischetti (minimo 10 e multipli di 10)
- Costanti allegati
- Assegno allegato n°
- Ho spedito l'importo a mezzo vaglia postale
- Ho versato l'importo sul CCP n° 19445204 intestato a J.soft - Milano
- Pagherò in contassegno al postino al ricevimento dei dischetti

Nome

Cognome

Via

CAP Città Prov.

Se richiesta fattura - codice fiscale

Data Firma

Offerta valida solo per l'Italia.

Dalla grande edicola Jackson

Tutto sul personal computer



PERSONAL O

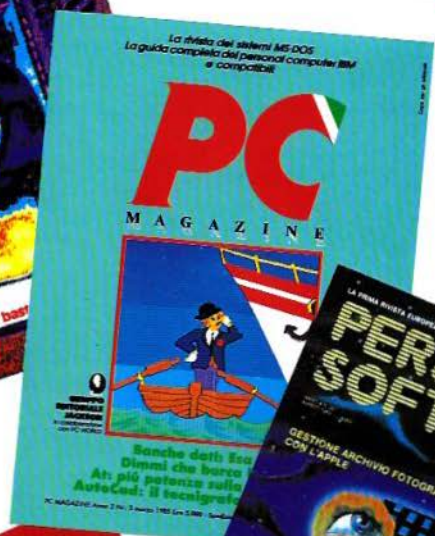
In questo numero:

L'architetto di fronte all'informatica

Unix per M 24

Il cancelliere nel cassetto

L'archivio tascabile, quando l'agenda non basta



PC MAGAZINE

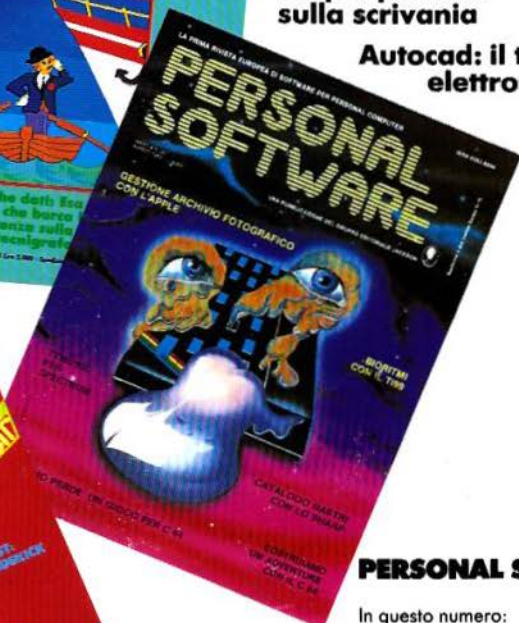
In questo numero:

Banche dati: è nostrana la più grande d'Europa

Il redditometro con VisiCalc

At: più potenza sulla scrivania

Autocad: il tecnigrafo elettronico



PERSONAL SOFTWARE

In questo numero:

Catalogo nastri per Sharp

L'ultimo perde con C64

TTmusic per Spectrum

Totocalcio con Apple



BIT

In questo numero:

Speciale Macintosh

HP-110 contro D.G.-One

Memotech MTX-512

Softtest: Super Window

Test: Sidekick



Personal-O/PC/Bit/Personal Software
sono pubblicazioni firmate:

GRUPPO EDITORIALE JACKSON

via Rosellini, 12-20124 Milano